Precision Agriculture Association New Zealand
TECHNOLOGY FOR SUSTAINABLE GROWTH
1st Asian-Australasian Conference on Precision Pastures and Livestock Farming

Fast object detection in pastoral landscapes using a multiple expert colour feature extreme learning machine

Edmund J. Sadgrove*, Greg Falzon, David Miron, David Lamb
Precision Agriculture Research Group, University of New England, NSW 2351 Australia

Abstract

Fast and accurate object detection is a desire of many vision-guided robotics based systems. Agriculture is an area where detection accuracy is often sacrificed for speed, especially in the pursuit of real time results. Pastoral landscapes are especially challenging with varying levels of complexity, as competing objects are rarely textually smooth or visibly different from surroundings. This study presents a machine learning algorithm designed for object detection called the Multiple Expert Colour Extreme Learning Machine (MEC-ELM). The MEC-ELM is a multiple expert implementation of a Colour Feature Extreme Learning Machine (CF-ELM). The CF-ELM is itself a modification of the Extreme Learning Machine (ELM) with a partially connected hidden layer and a fully connected output layer, taking 3 inputs. The inputs can be utilised by multiple colour systems, including, RGB, Y'UV and HSV. Colour inputs were chosen, as colour is not sensitive to adjustments in scale, size and location and provides information not available in the standard grey-scale ELM. In the MEC-ELM algorithm, feature extraction and classification techniques were implemented simultaneously making a fully functional object detection algorithm. The algorithm was tested on weed detection and cattle detection from a video feed, delivering 0.89 (cattle) to 0.98 (weeds) accuracy in tuning and a precision of 0.61 to 0.95 in testing, with classification times between 0.5s to 1s per frame. The algorithm has been designed with complex and unpredictable terrain in mind, making it an ideal application for agricultural or pastoral landscapes.

Background

Autonomous agricultural robotic based weed spraying or livestock monitoring requires the development of object detection software which can operate accurately and in real-time. Cascading or multiple expert approaches to robotic vision problems have shown good results in real time scenarios, in particular for facial and other prominent features in both flora and fauna applications (Berge et al. 2012; Wawerla et al. 2009; Uddin & Akhi, 2016). These applications however are often grey-scale discarding additional colour information and are limited to the detection of prominent features. Distinguishing weeds from pasture and stock from fauna can be challenging scenarios if such prominent features are unavailable. In agricultural environments slight variations in object colour, shape, orientation or texture may not be detected by a classification algorithm designed to distinguish prominent features. An Artificial Neural Networks (ANN) provides a solution for the detection of objects in such visually complex environments. ANNs look at an entire image at pixel-level; they are however, often limited to grey-scale values and require the use of feature extraction and scaling techniques. In contrast feature based algorithms can offer rapid processing and in the case of HAAR features (Mohamed et al. 2015) the innovative use of the summed area table (SAT). The SAT contains a sum of all values preceding each location in the image and allows fast transitions between image scales and the ability to find square based features within the image. The research in this paper has been motivated by the desire to merge the ANN and feature extraction architectures and hence, produce an algorithm that allows fast scaling and feature extraction in a complex environment. The proposed algorithm is called the multiple expert colour feature extreme learning machine (MEC-ELM). To allow colour object detection, the MEC-ELM will use four colour feature extreme learning machines (CF-ELM) (Sadgrove et al. 2017) each utilising SAT for input. The output of each CF-ELM (expert) is used to build a consensus object detector. The number of CF-ELM's can be varied in the MEC-ELM according to computational resources, four were selected in this scenario to match the number of cores in a standard personal computer.

Methods

The multiple expert colour feature extreme learning machine (MEC-ELM)

The MEC-ELM is a multiple expert implementation of the CF-ELM (Sadgrove et al. 2017). The CF-ELM is colour feature implementation of the extreme learning machine (ELM) (Guang-Bin Huang et al. 2006; Huang et al. 2015). The ELM (itself an ANN variant) processes on input per data sample and hence grey-scale imagery, in contrast, the CF-ELM has a partially connected hidden layer, with one set of neurons per colour band, making a three section hidden layer. Therefore the CF-ELM utilises a three band colour input, but since the CF-ELM has a fully connected output layer it also provides output weights similar to the ELM.

The Y'UV colour space

The Y'UV colour space was chosen for use with the CF-ELM, Y'UV was selected based on previous research (Podpora et al. 2014; Sadgrove et al. 2017) and produces superior results to other colour spaces including grey-scale. The selected colour space was defined by the international telecommunications union as ITU-R B.601 (International Telecommunications Union, 2015) and is more commonly known as YCbCr, where Y represents the luma value (light intensity/grey-scale) and CbCr are the blue-difference and red-difference chrominance values. Conversion from the RGB colour space can be calculated as a function of RGB (Al-Tairi et al. 2014).

The summed area table (SAT)

The SAT or integral image was made popular by the Viola and Jones object detection framework (Paul Viola & Michael Jones, 2004). The SAT allows real-time object detection using a HAAR feature base set and can be trained for a classification function using the AdaBoost algorithm (Wang, 2012). The HAAR features are groups of dark and light rectangular areas, each one of these areas can be matched to a feature within an image by adjusting a set of thresholds. The SAT can do this quickly, as it allows the sum of an area of an image to be returned in as little as four coordinates from the table. Each coordinate within the image becomes a sum of all pixel values preceding it, that is, above and to the left of the coordinate. The process for calculating each coordinate (x,y) of the image can be expressed (Facciolo et al. 2014),

$$I(x,y)=(x,y)+(x-1,y)+(x,y-1)-(x-1,y-1) \quad (1)$$

where I(x,y) is a single coordinate within the integral image and -1 refers to the previous coordinate in the dimensions x and y. Using this formula, the sum of all previous pixels above and to the left can be calculated. Starting at the top left hand corner of the image and processing through to the bottom right corner, the entire integral image can be processed from just one pass over the image. The sum of any rectangular area within an image can then be calculated with the values from just four coordinates. This can be expressed,

$$S(x,y,x-w,y-h)=(x,y)+(x-w,y-h)-(x-w,y)-(x,y-h) \quad (2)$$

where *S* refers to the sum of the rectangle within the image, x and y refer to the value stored at the bottom right coordinates of the rectangle and w and h refer to the width and height of the desired rectangle.

The algorithm

The algorithm is a multiple phase process designed for remote computer based classification based on the CF-ELM (Sadgrove et al. 2017), with an (i) initialisation phase: where the neural network is initialised into memory and the initial random weights stored, (ii) weight biasing phase: based on the CIW-ELM (Tapson et al. 2015), (iii) training phase: where the output weights of the ELM were determined with assistance from the C lapack library (Netlib.org, 2013) and each CF-ELM is trained on different images, (iv) tuning phase: which was used in combination with the training phase and finds a high accuracy CF-ELM and stores the weights off-line for future use and (v) testing phase: is depicted in figure 1. Similar to HAAR features, each image was divided equally into 25 to 100 sections (depending on the quality of testing results).

The training and tuning phases required each image to be converted into an integral image and from the resulting table, summed and averaged pixel values were sent as individual input values to each CF-ELM. The testing phase has multiple steps as displayed in figure 1. Starting from left to right, the video is first converted to frames, then individual frames are converted into 3 integral images (3II), one for each colour, then the 3II are fed to each CF-ELM in the MEC-ELM which reaches a consensus result via tallying. At multiple scales the image sections are tested and positive results recorded. If the tally reaches a set threshold, a square is draw in the location surrounding all sections tested in the area.

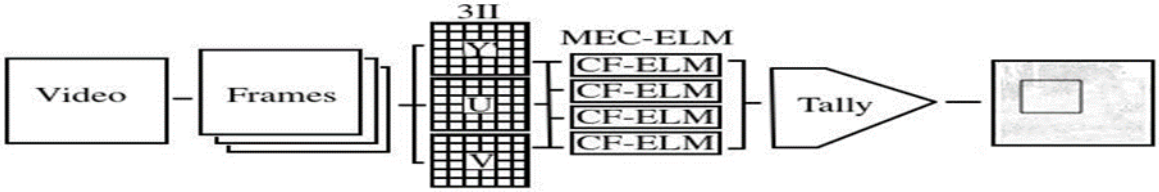

Figure 1. Classification algorithm from left to right, with conversion from video to frames to 3II and to the MEC-ELM and finally, tallying

Implementation

The MEC-ELM was programmed in the C programming language, C was chosen as variants of C are used on many embedded devices (Leskela et al. 2009) and it is generally faster and more efficient than other more commonly used programming languages. The C program was benchmarked on a Linux based system with a solid state drive, 16 gigabytes of RAM and a 4th Gen i7 mobile processor. The mobile processor simulated its potential use in a remote setting. All times in the results section were recorded using the clock_gettime function with the CLOCK_MONOTONIC option, available from the time.h library in C. All images were stored in the JPEG format, as JPEG is a common output for many remote camera interfaces and is also a useful format for both file storage and file transfer given size and transfer speed restrictions. All images were decompressed using the libjpeg library available in the C programming language; it is noteworthy that if speed and storage restrictions are not an issue, then the algorithm may benefit from a raw image format, as this avoids the decompression algorithms.

Datasets

Two datasets were used, one for weed detection and another for stock monitoring; the former being acquired over trials conducted on the Unive5rsity of New England SMART Farm. The weed detection dataset assesses the detection of weeds from a drone. The training/tuning dataset consisted of 688 images of *Cirsium Vulgare* (Bull Thistle). The thistle was photographed using a 10 mega-pixel hand held digital camera at a fixed distance of 2 metres to simulate low-level aerial detection. Object detection testing was conducted using 120 seconds of video (MP4 format, 4k with 4069 x 2178 colour pixel resolution) from a DJI Phantom drone flown low over a thistle infested paddock. The stock detection data set examines the use of a stationary camera for the monitoring of stock at a water point. Stationary video (Scoutguard SG860C surveillance camera, AVI format, 640 by 480 colour pixel resolution, 16 frames per second, 120 seconds) of Poll Hereford cattle at a creek crossing was used for algorithm testing based on 500 training/tuning frames.

Results

Detection results are on display in table 1, where column two indicates the number of sections used per input to the CF-ELMs and at how many image scales. Column three is the average accuracy per CF-ELM in the MEC-ELM, which was found while tuning with 50 images of the object and 50 images of surrounding landscape. Column four is the average time in seconds taken for each frame of the video feed. Column five is the true positive (TP), false positive (FP) and false negative (FN) results from all frames in the video feed. Columns six and seven are the detector precision and recall values (Olson & Delen 2008).

Table 1. MEC-ELM detections statistics for each dataset

Dataset	Sections / object	Accuracy in Tuning.	Time / Frame	Video Results	Precision	Recall
Bull Thistle	Sections:25 Scales: 2	0.98	1.001(s)	TP: 1206 FP: 68 FN: 153	0.95	0.89
Cattle	Sections:100 Scales: 2	0.89	0.445(s)	TP: 367 FP: 236 FN: 100	0.61	0.79

The results in table 1 indicate better results for thistle detection in all metrics, except times per frame. This was expected as the cattle video is of lower resolution than the thistle video. The left panel of figure 2 displays the results of the thistle detection video feed. The right panel of figure 2 demonstrates the detection of cattle, including a false detection of a reflection. Discarding reflections, the precision for cattle detection was approximately 0.78.

Figure 2. Thistle detection (left panel) and cattle detection (right panel) with square boxes indicating MEC-ELM detections. Note the false detection of the reflection in the right panel

Discussion

Multiple expert or cascading approaches to object detection in agriculture are generally limited to prominent features or rudimentary approaches (using just colour or dimensional attributes) (McCarthy et al. 2013; Vijayalaxmi et al. 2013; Wu et al. 2011), this research has presented an approach that utilises a neural network for the added complexity of the datasets and in particular the CF-ELM. The CF-ELMs allowed the classifier to use colour as an extra dimension, adopting the Y'UV colour space for better overall performance. The input of the MEC-ELM was configured to allow values from the output of the SAT for each colour space. This allowed the MEC-ELM to both extract features and classify objects in real time, with both datasets returning results between 0.5s and 1s per frame.

The SAT allowed the MEC-ELM to process image frames at multiple scales, with 2 scales used for each dataset, capturing both small and larger objects. This sectioning of the images into blocks of average colour may be considered a naive scaling technique (Culler et al. 1998), as it reduced the image into a lower resolution or a smaller set of pixels values. This implies, that the MEC-ELM, due in part to its cascading approach, has displayed an ability to classify objects at low resolutions and with further testing this may allow the video feed to operate at further reduced resolution, allowing even more rapid frame processing.

Identifying the cattle and weeds in real time could have multiple applications in agriculture, including cattle tracking and counting in the case of stationary cameras. Weed detection could be used for targeted weed spraying robotics, reducing the amount of chemical used, saving money while reducing the potential impact on the local environment. A drone system could identify areas for an unmanned ground vehicle (UGV) to spray, or alternatively a camera system could be mounted on an independently operating UGV.

The results reveal some potential weaknesses in the algorithm, with the cattle detection resulting in false positives from reflections in creek and a few from the colour of the soil being similar to the Poll Hereford. The thistle detection also suffered from some false positives, mostly in the edge regions of the frame, due to some blurring in these areas. To overcome these issues there will need to be further research into real time frame pre-processing prior to the MEC-ELM. This could include the removal of reflections, de-blurring or interest point detection. The MEC-ELM has demonstrated through quantifiable properties its potential use in both weed and cattle detection, in particular for use with mobile computer technology in agriculture.

Conclusion

The MEC-ELM has been tested as a real time feature extraction and classification algorithm in an agricultural setting. With testing on video of a drone flying over a paddock infested with bull thistle and a stationary camera trap observing cattle near a creek bed. The results show that the SAT can be used alongside a CF-ELM and deliver real time results. Weed detection achieved 0.98 accuracy in turning, 0.95 precision and 0.89 recall in testing at around 1.001s per frame and cattle detection achieved 0.89 accuracy in tuning, 0.61 precision and 0.78 recall in testing at 0.45s per frame. Algorithm benchmarking was performed on a mobile processor indicating that the MEC-ELM can be utilised in-field for the real-time detection of objects. Further research will include testing the algorithm in an embedded device and comparing results to other embedded detection methods in agriculture.

Acknowledgments

The authors would like to thank

Dr Paul Meek, NSW Department of Primary Industries for the provision of cattle surveillance data. Animal Ethics UNE AEC12-042, collected under scientific licence Sci Lic SL 100634. We would also like to thank Mr. Andrew Rieker of V-TOL Aerospace PTY Limited, for providing drone footage using their quadcopter and Mr. Paul Arnott, UNE SMART Farm, for access to paddocks for photography of weeds. Mr. E. Sadgrove is supported by an Australian Postgraduate Award.

References

Al-Tairi ZH, Rahmat RW, Saripan MI, Sulaiman PS 2014. Skin segmentation using YUV and RGB colour spaces. Journal of Information Processing Systems 10(2): 283–299.

Mohamed A, Issam A, Mohamed B, Abdellatif B 2015. Real-time detection of vehicles using the Haar-like features and artificial neuron networks. 73: 24–31.

Berge T, Goldberg S, Kaspersen K, Netland J 2012. Towards machine vision based site-specific weed management in cereals. Computers and Electronics in Agriculture 81: 79–86.

Culler DE, Singh JP, Gupta A 1998. Workload-driven evaluation. In Parallel Computer Architecture: A Hardware/software Approach (1st ed. 266 p.).

Olson DL, Delen D 2008. In advanced data mininig techniques (1st ed., 138 p.).

Facciolo G, Limare N, Meinhardt-Llopis E 2014. Integral images for block matching. 4: 344–369. http://doi.org/https://doi.org/10.5201/ipol.2014.57.

Huang G-B, Zhu Q-Y, Siew C-K 2006. Extreme learning machine: Theory and applications. Neurocomputing 70: 489–501.

Huang G, Huang G-B, Song S, You K 2015. Trends in extreme learning machines: A review. Neural Networks 61: 32–48.

International Telecommunications Union 2015. Studio encoding. Electronic Publication, Geneva.

Wawerla J, Marshall S, Mori G, Rothley K, Sabzmeydani P 2009. BearCam: automated wildlife monitoring at the Arctic Circle 20: 303–317. http://doi.org/DOI 10.1007/s00138-008-0128-0.

Leskela J, Nikula J, Salmela M 2009. OpenCL embedded profile prototype in mobile device. *In*: Signal Processing Systems. IEEE. http://doi.org/10.1109/SIPS.2009.5336267.

Mccarthy C, Rees S, Baillie C 2013. Preliminary evaluation of shape and colour image sensing for automated weed identification in sugarcane. International Sugar Journal 115(1376): 560–564.

Uddin MS, Akhi AY 2016. Horse detection using Haar like features 8(5): 415–418.

Netlib.org. 2013. The LAPACKE C Interface to LAPACK. Retrieved from http://www.netlib.org/lapack/lapacke.html.

Viola P, Jones M 2004. Robust real-time object detection. International Journal of Computer Vision 57(2): 137–154.

Podpora M, Korbas GP, Kawala-Janik A 2014. YUV vs RGB – Choosing a colour space for human-machine interaction. Federated Conference on Computer Science and Information Systems 3: 29–34. http://doi.org/10.15439/2014F206.

Wang R 2012. AdaBoost for feature selection, classification and its relation with SVM: A review. 25: 800–807.

Sadgrove EJ, Falzon G, Miron D, Lamb D 2017. Fast object detection in pastoral landscapes using a Colour Feature Extreme Learning Machine. Computers and Electronics in Agriculture 139: 204–212.

Tapson J, de Chazal P, van Schaik A 2015. Explicit computation of input weights in extreme learning machines. *In*: International Conference on Extreme Learning Machines, Switzerland. 1: 41–49.

Vijayalaxmi PB, Putta R, Shinde G, Lohani P 2013. Object detection using image processing for an industrial robot. International Journal of Advanced Computational Engineering and Networking 1(7): 21–26.

Wu X, Xu W, Song Y, Cai M 2011. A detection method of weed in wheat field on machine vision. Procedia Engineering 15:1998–2003.