# Learning Image-based Contaminant Detection in Wool Fleece from Noisy Annotations

Timothy Patten[1,2][0000−0003−1139−9451], Alen Alempijevic[1][0000−0002−1769−8041], and Robert Fitch[1][0000−0003−2215−5188]

[1] Robotics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia
firstname.lastname@uts.edu.au

[2] Automation and Control Institute, Faculty of Electrical Engineering and Information Technology, TU Wien, Vienna 1040, Austria

**Abstract.** This paper addresses the problem of detecting natural contaminants in freshly shorn wool fleece in RGB images using deep learning-based semantic segmentation. The challenge of inconsistent annotation is overcome by learning the probability of contamination as opposed to a discrete class. From the continuous value predictions, contaminated regions can be extracted by selectively thresholding on the probability of contamination. Furthermore, the imbalance of the class distributions is accounted for by adaptively weighting each pixel's contribution to the loss function. Results show that the adaptive weight improves the prediction accuracy and overall outperforms learning an approximated representation by quantising the distributions.

**Keywords:** Wool · Contaminant detection · Semantic segmentation · Deep learning · Annotation noise.

## 1 Introduction

Wool handling is a highly manual and repetitive task. A major bottleneck in this activity is the identification of contaminants [6] that must be removed by hand, known as *skirting* (Fig. 1). Automation of this process can improve both efficiency and accuracy, which is anticipated to increase the productivity and thus translate to significant savings in cost for the wool growers.

A vision system that identifies contaminants in wool is a crucial aspect of the automation process. The system should capture images and specify the regions that contain contaminants and those that have clean wool. Previously studies on contaminant detection in wool from monocular cameras [25, 18] consider non-organic substances such as fibres from packing material, fertiliser bags or hay bale twine. These methods use thresholding to identify image pixels with colour properties different to the background wool. The *natural* contaminants that are removed from fresh wool during skirting, such as urine-stain, dung, pigmentation/medullation and vegetable matter, are less distinguishable by their colour and, therefore, prior work for an on-farm system is limited.

Fig. 1: The manual skirting process of contaminated wool fleece [1]. Left: Two wool handlers move along the edge of the fleece. Right: Close up of the hand position and gripping.

Machine learning, in particular deep learning, has profoundly improved the performance of visual perception [8] for a broad range of applications such as recognition, classification and segmentation [20]. Recently, deep learning has been applied in the related task of foreign fibre detection in cotton [21, 22]. We conjecture that the application of deep learning to contaminant detection in wool fleece will not only deliver highly accurate results but also generalise to more difficult scenarios, where colour differentiation, as in [25, 18], is unsuited.

This work addresses the challenge of learning to identify natural contaminants in wool fleeces from RGB images. A strong prerequisite for this task is access to a large, annotated dataset, which constitutes pixel-level annotation of the contaminated wool. Unfortunately, annotation may be inconsistent due to the difficulty of identifying contamination in images. Furthermore, some contaminants are continuous in nature – contamination is in the spectrum from "light" to "heavy". As such, annotators may disagree on what parts of the fleece should be removed. The end result is that multiple annotations are presented to the learning algorithm from which a meaningful output should be derived; see Fig. 2. Consequently, our system learns to predict the continuous blending between the classes based on noisy pixel labels. More specifically, a deep learning architecture for pixel-wise classification is employed to predict the probabilities of each class. We propose to weight the influence of each pixel in the loss function according to the frequency of the distribution in the dataset by extending the formulation in [13].

We report the performance of deep learning-based semantic segmentation on newly collected and annotated datasets that represent two scenarios: (1) Delineation around the edge of wool fleeces that separates the portion that is skirted and (2) Bathurst burr (a species of weed) contamination on fleece off-cuts. In comparison to predicting a quantised set of classes, the probabilistic output is more general. We show how adjustment on the contaminant threshold can be set to promote more aggressive or conservative contaminant removal, which is highly useful in industry as external factors, such as the daily market value or lot size, contribute to the degree at which wool handlers remove contaminants.

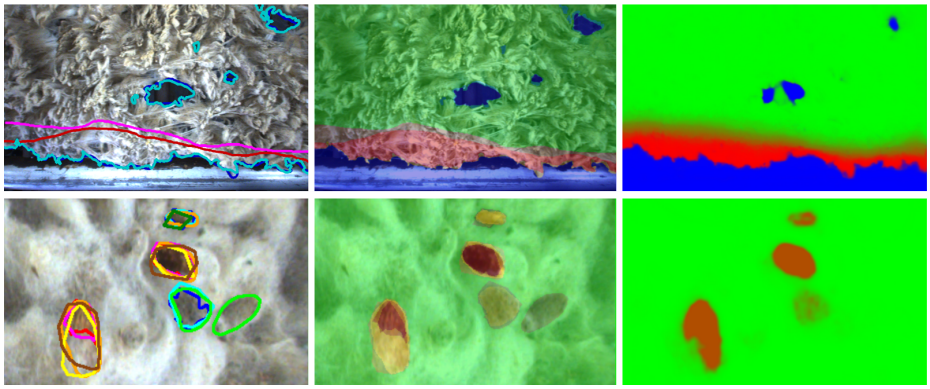

Fig. 2: Examples of wool contamination and annotation. *Dubbo* dataset on top row and *Bathurst burr* dataset on bottom row. Left: Inconsistent annotations of contaminated wool, background and vegetable matter. Middle: Discrete class representation from the combinations of the annotations. Right: Probabilistic prediction of each class encoded to RGB.

The remainder of this paper is organised as follows. Sec. 2 discusses related work. In Sec. 3, we present our approach for predicting class probabilities from noisy labels. Sec. 4 introduces the new datasets of wool contamination. Sec. 5 presents the experiments and finally in Sec. 6 we conclude the paper.

## 2 Related Work

### 2.1 Contaminant Detection in Wool

Detection of contaminants in wool from vision sensors has been a topic of research in wool technology for decades [4]. In this early work, an NIR camera observes the wool and then PCA with soft independent modelling of class analogies is applied to discriminate between polymeric material (polyethylene and polypropylene) and contaminant-free wool. More recently, Zhang et al. [25] present a system that uses RGB cameras, in which the polypropylene contaminants are identified through global and local adaptive threshold in the RGB and HSV colour spaces. This vision system is integrated with a mechanical system in [24] to additionally remove and sort the contaminants in real time. Similarly, Su et al. [18] develop an online system to detect and remove contaminants in wool. As in [25, 24], thresholding on the RGB values identifies the contaminants.

While these systems show strong results, they focus on detecting contaminants that are vastly different to the wool. The approaches do not detect natural contaminants, which appear as stains or more subtle colour/textural changes compared to plastic material. A more promising direction is the use of modern deep learning techniques, which has been applied to the related task of detecting foreign fibres in cotton [16, 21, 22]. Indeed, this task has attracted significant attention that dedicated datasets have been compiled for developing and training learning algorithms [12].

### 2.2 Semantic Segmentation

Within the computer vision community, the task of assigning a class label to every pixel in the image is known as *semantic segmentation*. This differs from classification for which an entire image is assigned a single class label. In the context of deep learning, seminal work by Long et al. [10] adapted existing convolutional neural networks (CNNs) for classification to the task of semantic segmentation by converting them to fully convolutional architectures through the replacement of the classification layer(s) with an upsampling convolutional layer. Ronnenberg et al. [13] introduce U-Net, which extends [10] by mirroring the convolution structure in the upsampling stage with consecutive convolutional layers. The larger number of feature channels in the decoder, as compared to [10], propagates contextual information to higher resolution layers, yielding more precise segmentation with fewer training images. A plethora of adaptations and modifications now exist, which is motivated by the broad range of applications that this technique can be applied, e.g., autonomous driving [7, 15].

Supervised learning from noisy pixel-wise labels is a growing trend because it is often difficult to avoid annotation error in complex, real-world data [5, 17]. Zheng and Yang [26] show that incorporating predicted uncertainty in the optimisation procedure leads to better domain adaptation when learning from pseudo labels. In similar work, noisy pre-segmentation masks in the target domain are refined by a label cleaning network that is trained jointly with the segmentation network using the same feature encodings [9]. Similar to our work, the problem of noise at boundaries (i.e., label transition) due to poor image resolution or annotator error is addressed in [2]. By introducing a pixel-weight that is determined by the pixel's distance to its nearest pixel of another class, the loss function is less affected by potential incorrectly labelled pixels. Cheng et al. [3] train a GAN to both revise spatially noisy labelling and output a label weight that, similar to [2], reduces the influence of untrustworthy samples. This method, however, requires access to the underlying ground truth during training in order to learn the mapping between clean and noisy samples.

## 3 Semantic Segmentation from Noisy Annotations

### 3.1 Problem Definition

Classical semantic segmentation assumes a dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ that consists of images $\mathcal{X}$ with corresponding labels $\mathcal{Y}$. Each pixel $\mathbf{x} \in \Omega$ has a corresponding unique label $y_{\mathbf{x}}$ belonging to one of the known set of classes $\mathcal{C} = \{1, \ldots, C\}$. The task during inference is to predict the label $y_{\mathbf{x}}$ for each $\mathbf{x}$ for any test image.

In this work, labelled images are provided by $N$ annotators, in other words, $\tilde{\mathcal{D}} = \{\mathcal{X}, \mathcal{Y}^1, ..., \mathcal{Y}^N\}$. Each $\mathbf{x}$ is assigned a set of potentially different labels, thus no single class can be assumed. Instead, each pixel has a corresponding probability distribution over the classes $\mathcal{C}$. Formally, the dataset consists of images where each $\mathbf{x} \in \Omega$ has an associated probability distribution $p(\mathbf{x}) = F(y^1_{\mathbf{x}}, ..., y^N_{\mathbf{x}})$ where

$F : \mathbb{Z}^{C\times C} \rightarrow \mathbb{R}^C$ maps the set of annotated labels to a probability distribution over $\mathcal{C}$ such that

$$\sum_{c\in\mathcal{C}} p(x_c) = 1, \tag{1}$$

is satisfied where $x_c \in \mathbf{x}$ are the individual class probability values. The task during inference is to predict the probability $p(\mathbf{x})$ for each $\mathbf{x}$ for any test image.

### 3.2 Learning Probabilistic Output

Pixel-wise prediction using deep convolutional networks was first introduced by Long et al. [10]. The fully convolutional network learns a pixel-to-pixel mapping through an encoder (the feature extraction layers of a classification CNN) and a decoder to upsample the feature map to the original input resolution. The output is a class prediction for each pixel and the parameters are optimised according to a classification loss applied over every pixel in the input image. Numerous modifications and enhancements have been applied to extend this basic principle. A good discussion is given in [11].

Our work learns to predict a distribution instead of a discrete class, which is trivially achieved by using the *softmax* activation as final layer. The activation function outputs the value of each class $z_c \in \mathbf{z}$ in the range $[0, 1]$. Eq. (1) is satisfied due to the normalisation in the denominator

$$\sigma(\mathbf{z})_c = \frac{e^{z_c}}{\sum_{c\in\mathcal{C}} e^{z_c}}. \tag{2}$$

A variety of loss functions for real value regression can be applied to minimise the difference between the input probability distributions and the corresponding predictions. Prior work for classical semantic segmentation show that the issue of class imbalance should be addressed by assigning a weight to each class [13, 19] so that classes represented by very few samples have more contribution during learning. As such, the more common classes do not dominate the optimisation, enabling less common classes to also be precisely learned. For the discrete class scenario, the inverse of the volume of each class is used as the weight. For the probabilistic scenario, computing the weight is less straight forward because it is uncountable.

We propose a weighting scheme based on the frequency of the probability distribution for each pixel in the ground truth images. For each pixel $\mathbf{x}$, the weight is computed by

$$w(\mathbf{x}) = \prod_{c\in\mathcal{C}} f_c \left(p(x_c)\right)^{-\gamma}, \tag{3}$$

where $f_c : \mathbb{R} \rightarrow \mathbb{R}$ maps a pixel's ground truth probability of class $c$ to a frequency value and $\gamma \geq 0$ is a constant. The functions $f_c$ are approximated by fitting a Gaussian process (GP) to the histogram of the counts per probability value. We use 10 bins and a GP with the Matérn kernel with length scale 1.0 and smoothing factor 1.5.

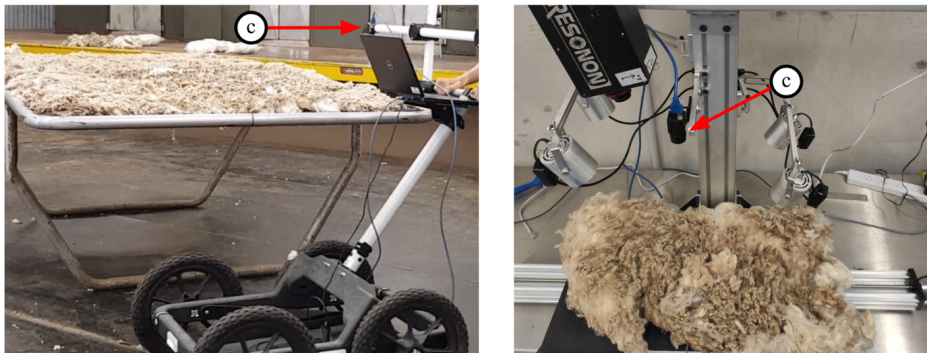


Fig. 3: Acquisition of wool datasets (camera indicated by **c**). Left: Camera mounted on trolley that is pushed parallel to freshly shorn wool fleeces. Right: Plate mounted on linear actuator that moves wool samples below static camera.

The weights are applied to each pixel when computing the loss during training. For example, the pixel-wise $L_2$ loss (i.e., mean squared error)

$$L_2(\Omega) = \sum_{\mathbf{x}\in\Omega} \frac{1}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} \left(\hat{p}(x_c) - p(x_c)\right)^2, \tag{4}$$

becomes

$$L_2^w(\Omega) = \sum_{\mathbf{x}\in\Omega} \frac{w(\mathbf{x})}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} \left(\hat{p}(x_c) - p(x_c)\right)^2, \tag{5}$$

where $\hat{p}(x_c)$ is the predicted probability for class $c$ of pixel $\mathbf{x} \in \Omega$. Similar modifications can be applied to other losses such as the $L_1$ or Huber loss.

## 4 Wool Datasets

Datasets of RGB images of wool fleece were collected to evaluate our proposed work. A Flir Blackfly S USB3 RGB camera[3] was used with a resolution of 1440×1080 pixels. Example images with annotation are provided in Fig. 2.

The *Dubbo* dataset consists of images of freshly shorn Merino fleece collected on site in Dubbo, NSW, Australia. The camera was mounted on a trolley that was pushed by a human operator along the edge of the skirting table; see Fig. 3 (left). Scanning was conducted at a slow pace to minimise motion blur. In total, five fleeces were scanned four times, i.e., once per edge. For each scan, images were hand selected such that visual overlap was minimised, resulting in a dataset consisting of 126 images. The skirting lines were annotated by two expert wool handlers. As discussed, the annotations do not necessarily agree due to differing opinions and the difficulty to identify the boundary between the "good" and contaminated wool in the images. In addition, the background was also annotated. This was performed by non-wool handlers as domain experience was not

[3] https://www.flir.com.au/products/blackfly-s-usb3/

required. However, disagreement also exists in this annotation due to human error or ambiguity, e.g., thin wool that appears translucent over the background. The final result was a set of masks, from each annotator, that depicts the regions of the images that are either good wool (above the skirting line) or background (below the background line). The remaining class of contaminated wool is derived from the unlabelled pixels of the merged masks.

The *Bathurst burr* dataset consists of images of Merino fleece off-cuts. Each off-cut was placed on the rig in Fig. 3 (right), which moves the wool underneath the downward-facing statically mounted camera. Ten pieces of wool were scanned and for each scan, three images were hand selected to minimise image overlap. Due to the small number of images, 320×320 crops were extracted from the full images to yield a total of 380 images. In this dataset, the wool pieces are taken from different parts of the fleece and contain heavy Bathurst burr contamination. Each annotator was requested to label the instances of burr that they were confident about as well as those that they were uncertain about, by assigning a different label. A total of five people performed the annotation.

The probabilistic annotation was obtained by merging the masks of each image. Then, for each pixel, the number of class votes were counted and normalised to obtain the probability distribution. For the *Bathurst burr* annotation, each pixel assigned the uncertain label contributed a vote of 0.5.

Due to the small size of the datasets and to avoid samples from the same fleece or piece of wool appearing in both the training and test sets, we provide multiple splits for k-fold cross-validation. For the *Dubbo* dataset, for each fold, one fleece is held out for testing while the remaining four fleece are used for training. For the *Bathurst burr* dataset, pairs of scans, i.e., 1-2, 3-4, ..., 9-10, are held out for testing and all remaining scans are used for training.

## 5 Experiments

### 5.1 Implementation Details

This work is implemented in PyTorch using the segmentation models library [23]. We use the U-Net architecture [13] with MobileNet v2 encoder [14] as this setup is lightweight, thus better for our datasets with small numbers of training images. Models are pre-trained on ImageNet then trained on the wool datasets with a batch size of 4 for 30 epochs with an initial learning rate of $10^{-3}$ that is divided by 10 at epoch 15 and 25. All layers use *ReLu* activation except the last layer that uses *softmax* activation. Batch normalisation is applied between the convolutional and activation layers in the decoder. Images in the *Dubbo* dataset are input to the network with the dimension 640×640 by padding the shorter dimension and resizing. Images in the *Bathurst burr* dataset are input at the patch resolution of 320×320. All training and testing is performed on an NVIDIA RTX 2080 Ti.

Table 1: Mean KL divergence for subsets of pixels for varying values of $\gamma$ in Eq. (3) applied to $L_2$ loss. $\gamma = 0.0$ is equivalent to the unweighted loss in Eq. (4).

| Dataset | Dubbo | | | | Bathurst burr | | | |
|---|---|---|---|---|---|---|---|---|
| $\gamma$ | 0.0 | 0.1 | 0.5 | 1.0 | 0.0 | 0.1 | 0.5 | 1.0 |
| good | **0.007** | **0.007** | 0.012 | 0.020 | **0.001** | **0.001** | 0.002 | 0.004 |
| cont. | 0.044 | **0.039** | 0.072 | 0.148 | 0.196 | 0.093 | 0.086 | **0.075** |
| mix | 0.072 | 0.057 | 0.028 | **0.014** | 0.045 | 0.043 | 0.034 | **0.025** |
| all | 0.015 | **0.013** | 0.020 | 0.045 | 0.004 | **0.003** | **0.003** | 0.005 |

### 5.2 Results and Discussion

**Weighted Loss Function** The performance of the proposed weighting scheme for various values of $\gamma$ in Eq. (3) applied to the $L_2$ loss in Eq. (5) is provided in Tab. 1. We report the mean Kullback–Leibler (KL) divergence for fully-uncontaminated (*good*), fully-contaminated (*cont.*) and partially-contaminated (*mix*) pixels. The bottom row shows overall performance (*all*). Note that fully-uncontaminated pixels are significantly more common than fully- or partially-contaminated pixels. For both datasets, increasing $\gamma$ improves the predictions of the contaminated and mixed pixels at a sacrifice in performance on the good pixels. This is expected because larger $\gamma$ means a more extreme difference in the weights between the dominant and less common distributions. With $\gamma = 1.0$, although the gains on the contaminated and mixed pixels are significant, the reduction of the more present good pixels results in worse overall performance compared to no weighting scheme (i.e., $\gamma = 0.0$). A smaller value of $\gamma$ is less aggressive and strikes a balance, which achieves best performance for *all* pixels.

**Comparison to Classification:** We compare the performance of learning to regress probabilities as opposed to learning a discrete set of quantised classes on the *Bathurst burr* dataset. For discrete learning, we create sets of classes that quantise the probabilities to different resolutions. The dice loss is employed for learning classification. As shown in Fig. 4, when considering all pixels, the performance between continuous and discrete learning is highly similar. However, for the pixels that have a mixed class probability distribution, there is a notable difference. Discrete learning has worst performance when the number of classes is small. This is because the resolution is insufficient to predict the continuous values. As the number of classes increases, the performance improves. However, with even more classes, the performance begins to decrease because the classification task is more difficult. Overall, directly regressing the probabilities performs better than learning any discrete representation.

**Flexible Contaminant Detection:** An advantage of learning a probabilistic representation is that the contamination boundary can be flexibly extracted from a single trained model. An adjustable threshold can be applied such that a different skirting line is generated based on the contour in the predicted probability map as shown in Fig. 5. This allows more conservative or aggressive contaminant removal. With a low probability of contamination, e.g., 25%, the predicted skirting line follows the region where all annotators agree on contamination. On

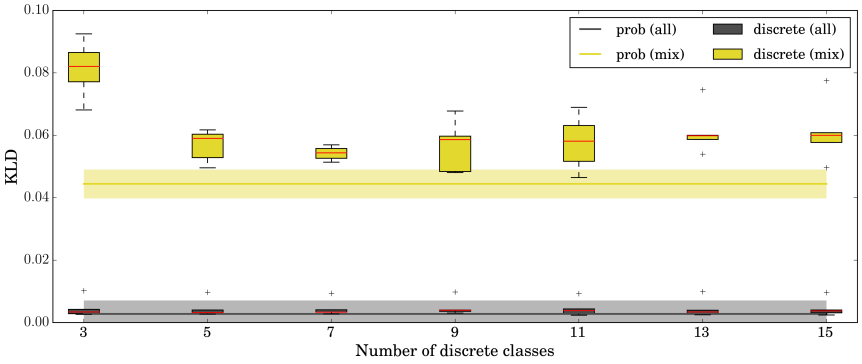


Fig. 4: Performance of learning continuous probabilities by regression in comparison to discretising and learning by classification on the *Bathurst burr* dataset. Boxes and error region derived from results of different splits.

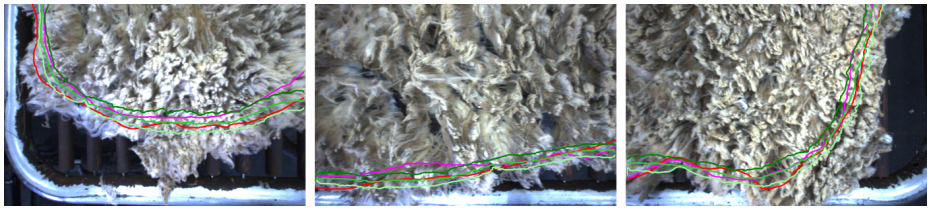

Fig. 5: Predicted skirting lines for contamination probability of 25, 50 and 75% in increasing shade of green. Annotated skirting lines in red and magenta.

the other hand, for a higher contamination probability, e.g., 75%, the predicted skirting line follows the boundary where at least one annotator indicates contamination. In between the extremes, e.g. 50%, the predicted skirting line is approximately between the annotated lines.

## 6 Conclusion

This paper analysed the capability of deep learning-based semantic segmentation for detecting natural contaminants in wool fleece from RGB images. To account for the ambiguous annotation, we learned probabilistic outputs rather than discrete class labels. The imbalance of the probability distributions was addressed by a weighting scheme to bolster the loss function to the less occurring probabilities. Our experiments on two new wool datasets showed that the weighting scheme improves accuracy and that learning probabilities is more accurate than learning discrete representations.

In future work we will expand on the datasets as well as incorporate more contaminants such as dermatitis, wool rot and other varieties of vegetable matter. We also plan to investigate the capability of detecting contaminants in hyperspectral images. Finally, the vision system will be integrated with a mechanical system to physically remove contaminants in the automated skirting process.

## Acknowledgement

This work is supported by funding from Australian Wool Innovation Limited, grant number ON-00713. The authors thank Craig French and Penny Clout for their expert annotation.

## References

1. Australian Wool Innovation Limited: https://www.wool.com/people/shearing-and-woolhandling/training-resources/ (2015)
2. Bressan, P.O., Junior, J.M., Martins, J.A.C., Gonçalves, D.N., Freitas, D.M., Osco, L.P., de Andrade Silva, J., Luo, Z., Li, J., Garcia, R.C., Gonçalves, W.N.: Semantic segmentation with labeling uncertainty and class imbalance. arXiv **2102.04566** (2021)
3. Cheng, G., Ji, H., Tian, Y.: Walking on two legs: Learning image segmentation with noisy labels. In: Proc. of UAI. pp. 330–339 (2020)
4. Church, J., O'Neill, J.: The detection of polymeric contaminants in loose scoured wool. Vib. Spectrosc. **19**(2), 285–293 (1999)
5. Han, B., Yao, Q., Liu, T., Niu, G., Tsang, I.W., Kwok, J.T., Sugiyama, M.: A survey of label-noise representation learning: Past, present and future. arXiv **2011.04406** (2020)
6. Hansford, K.: Contamination. University of New England, WOOL472:Wool biology and metrology (coursenotes) (2012), https://www.woolwise.com/wp-content/uploads/2017/07/WOOL-472-572-12-T-12.pdf
7. Kaymak, Ç., Uçar, A.: A brief survey and an application of semantic image segmentation for autonomous driving. In: Handbook of Deep Learning Applications. Smart Innovation, Systems and Technologies. vol. 136, pp. 161–200 (2019)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proc. of NeurIPS. p. 1097–1105 (2012)
9. Li, Y., Jia, L., Wang, Z., Qian, Y., Qiao, H.: Un-supervised and semi-supervised hand segmentation in egocentric images with noisy label learning. Neurocomputing **334**, 11–24 (2019)
10. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proc. of IEEE CVPR. pp. 3431–3440 (2015)
11. Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: A survey. IEEE Trans. Pattern Anal. Mach. Intell. (2021), [early access]
12. Pelletier, M.G., Holt, G.A., Wanjura, J.D.: A plastic contamination image dataset for deep learning model development and training. AgriEngineering **2**(2), 317–321 (2020)
13. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Proc. of MICCAI. pp. 234–241 (2015)
14. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: Proc. of IEEE CVPR. pp. 4510–4520 (2018)
15. Siam, M., Elkerdawy, S., Jagersand, M., Yogamani, S.: Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In: Proc. of IEEE ITSC. pp. 1–8 (2017)

16. Siddaiah, M., Prasad, N.R., Lieberman, M.A., Hughs, S.E.: Identification of trash types and computation of trash content in ginned cotton using soft computing techniques. In: Proc. of MWSCAS. pp. 547–550 (1999)
17. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: A survey. arXiv **2007.08199** (2021)
18. Su, Z., Tian, G.Y., Gao, C.: A machine vision system for on-line removal of contaminants in wool. Mechatronics **16**(5), 243–247 (2006)
19. Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. pp. 240–248 (2017)
20. Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E., Andina, D.: Deep learning for computer vision: A brief review. Comput. Intell. Neurosci. **2018**(7068349) (2018)
21. Wei, W., Deng, D., Zeng, L., Zhang, C., Shi, W.: Classification of foreign fibers using deep learning and its implementation on embedded system. Int. J. Adv. Robot. Syst. **16**(4), 1–20 (2019)
22. Wei, W., Zhang, C., Deng, D.: Content estimation of foreign fibers cotton based on deep learning. Electronics **9**(11) (2020)
23. Yakubovskiy, P.: Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch (2020)
24. Zhang, L., Dehghani, A., Zhenwei Su, King, T., Greenwood, B., Levesley, M.: Development of a mechatronic sorting system for removing contaminants from wool. IEEE/ASME Trans. Mechatronics **10**(3), 297–304 (2005)
25. Zhang, L., Dehghani, A., Su, Z., King, T., Greenwood, B., Levesley, M.: Real-time automated visual inspection system for contaminant removal from wool. Real-Time Imaging **11**(4), 257–269 (2005)
26. Zheng, Z., Yang, Y.: Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. Int. J. Comput. Vis. **129**(4), 1106–1120 (2021)