33rd European Modeling & Simulation Symposium
18th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0029 ISBN 978-88-85741-57-7
doi: 10.46354/i3m.2021.emss.042

Multi-Resolution Localization of Individual Logs in Wooden Piles Utilizing YOLO with Tiling on Client/Server Architectures

Christoph Praschl[1,*], Philipp Auersperg-Castell[3,4], Brigitte Forster-Heinlein[4] and Gerald Adam Zwettler[1,2]

[1]Research Group Advanced Information Systems and Technology, Research and Development Department, University of Applied Sciences Upper Austria, Softwarepark 11, Hagenberg, 4232, Austria
[2]Department of Software Engineering, School of Informatics, Communications and Media, University of Applied Sciences Upper Austria, Softwarepark 11, Hagenberg, 4232, Austria
[3]Bluedynamics Auersperg-Castell KG, Kritzing 31, 4785, Freinberg, Austria
[4]Faculty of Computer Science and Mathematics, University of Passau, Innstraße 43, Passau, 94032, Germany

[*]Corresponding author. Email address: christoph.praschl@fh-hagenberg.at

Abstract

In industrial domains with time and cost intensive manual or semi-automated inspection the demand for automation is high. Utilizing state of the art deep learning models for localization in vision-based domains such as wood log analysis, the precision can be increased while reducing the demand for manual inspection. In this paper a YOLO network is trained on wood log images to allow for detection of single wood piles in images with hundreds and thousands instances. Due to the high variability in scale and large amount of wood logs within the images, common YOLO architectures are not applicable. Thus, tiling is necessitated to implicitly form a multi-resolution image pyramid. Due to lack in training data, besides common data augmentation modelling of different seasonal and weather conditions is applied. The wood log detection process can be run on a client/server architecture to allow for both, preview and refined results. Evaluation on real-world data sets shows an log detection accuracy of 82,9% utilizing a tiny YOLO model and 94,1% with a fully connected YOLO model, respectively.

Keywords: Multi-resolution YOLO; Data Augmentation; Image Tiling; Wood Log Analysis

1. Introduction

With the recent improvements in computer vision due to availability of good deep learning paradigms, machine learning frameworks and improved GPU hardware, the automated vision-based measurement becomes feasible in many industrial areas. Thus, as an aspect of digitization and industry 4.0, more and more production processes are now performed in a semi-automated way, e.g. monitoring the flames on the skelp production line as described by Chen et al. (2020), human-machine cooperation in manufacturing as shown by Paredes-Astudillo et al. (2020) or utilization of autonomous vehicles in factory storage depots as presented by Flämig (2016). But even the hardware of consumer smart phones is nowadays able to perform

computational intensive augmented reality and computer vision tasks. Thus, a digital ruler with or without depth data to measure one's room in the sense of augmented reality paradigms has already turned into reality as described by Reitinger et al. (2005), Schmucker et al. (2019) and Murata et al. (2018).

However, in forestry industry, for the domain of wood pile trading, key aspects for the price such as cross section of the logs, quality and type of the wood are most of the time still assessed in a manual way. As this is a very time consuming process, digitization is a key factor in cost reduction. Utilizing a smart phone app, the log front faces can be automatically detected and segmented besides precisely quantifying both, the wood type and the quality in an objective and reproducible way.

1.1. State of the Art

The task of localization has been addressed with shape based strategies in the last decades. Daugman (2006) uses Hough circles on edge representations and is able to detect robustly the iris within a human eye, while Ballard (1981) utilizes the generalized Hough transformation (GHT) to get arbitrary shapes within an image invariant of pose and scale. In contrast, other localization approaches model the local gradient characteristics. With both, binary local patterns (BLP) as described by Ojala et al. (1994) and Histogram of Oriented Gradients (HOG) as shown by Dalal and Triggs (2005), simple shape detectors can be constructed. To boost a collection of rather weak local convolution-based detectors, Haar Cascades as used for face detection by Viola and Jones (2001) are utilized in a broad field of applications and can be seen as the shallow and sequential ancestor of modern convolutional neural networks as analysed by Cengil and Çinar (2017).

With the convolution as key image processing function for CNNs, various network and architecture types have been recently introduced. While modern machine learning frameworks cover most of the requirements for numeric optimization, heuristic search as well as data augmentation, the strategy for scale and pose variance is a key differentiating factor. Thereby, YOLO (you only look once) claims to process the image in one run only, implicitly handling image pyramid scale as introduced by Redmon et al. (2015). Gold standard frameworks such as Mask RCNN as presented by He et al. (2017) or YOLAct introduced by Bolya et al. (2019) combine segmentation and bounding-box based localization for general object detection and instance segmentation tasks. While YOLAct is capable of being applied in real-time, this model has problems with too many objects in one spot, which is critical for the area of application addressed by this paper. In contrast to that, Mask RCNN is not real-time capable and therefore is unsuitable for this area of application with demand for first result approximation to be calculated on smartphones or tablets as target hardware.

In comparison to that, frameworks such as YOLO only result in classified bounding boxes without any instance segmentation, with the advantage of real-time capabilities and lower requirements according to computation power and memory. This allows YOLO to be also used in outdoor mobile applications as on smartphones, where a high-performance hardware is not available as described by Mahmoud et al. (2020) and Bochkovskiy et al. (2020). The performance advantage can be further increased using adapted YOLO models as the scaled YOLOv4-tiny introduced by Wang et al. (2020). With many different yet promising deep learning network architectures and paradigms available, the commonly hand-designed adaption to new domains can be performed in an automated way too, utilizing neural architecture search(NAS) with AutoML for DL applications if a sufficient amount of training data is present as shown by Elsken et al. (2019).

1.2. Related Work

When adapting deep learning approaches to new application domains, the availability of a sufficient number of representative training data samples together with accurate ground-truth is a key requirement that is hardly ever met. A general chicken-egg-problem arises as the training data needs to be prepared in a very time-consuming semi-automated way first to allow for subsequent training of rough deep learning models then. To cover the demand for training data, several strategies have been presented in the past.

With the utilization of weaker classification approaches such as Haar Cascades introduced by Viola and Jones (2001) the demand for training data is significantly reduced, thus allowing first rough results in detection as analysed by Auersperg-Castell (2018) for wood logs. Besides, GANs can be utilized to synthesize images and thus to enrich the available number of training samples as shown by Zwettler et al. (2020) in the medical domain. With a first weakly-trained deep learning model available, the amount of training data can then be iteratively expanded by visual inspecting and post-processing the DL result. For post processing, Graph cut segmentation can be applied in a semi-automated way as introduced by Zwettler et al. (2021) or the DL model is trained together with a priori defined marker for manual adjustment as published by Sakinis et al. (2019). Furthermore, transfer learning can help to conquer the demand for training data too by gaining well-trained weights from similar application domains first, prior to adjusting and refining the model to the particular target domain, e.g. reflective elevator cabins, as shown by Reithmeier et al. (2021).

1.3. Multi-Resolution Localization of Individual Logs in Wooden Piles Utilizing YOLO with Tiling

In this paper a pragmatic multi-resolution approach for localization is presented that allows for processing on small data samples and also features a client-server architecture to allow for the trade-off between accuracy and processing on the fly. With YOLO applied to a multi resolution pyramid, logs can get precisely detected even in piles with hundreds and thousands of wood logs. The research questions to be addressed in this research work are as follows:

- Can YOLO be used in a multi-resolution approach to allow for localization of small wood logs?
- Can consumer smartphones be utilized to achieve preview results in real-time that are refined on a server for high-quality results?
- Can lack in seasonal wood log characteristics (*snow, mud,...*) and weather conditions for the acquired images be compensated by advanced data augmentation strategies?

2. Material

In Austria there are two main kinds of coniferous forests, such ones with spruces and firs, and the second group with douglas firs, jaws and larchs. In addition to the actual tree type, every log can be classified based on different quality features such as cracks, beetle infestation, or colour differences. Especially the colour features are not the same over all types of trees and for this have to been considered associated with the tree type. For example a red discoloration of the log may be a bad indicator for most types of trees but is a typical feature of douglas fir logs. In the context of this project we consider four classes of wood qualities that are handled by sawmills in the Austrian wood market. For this reason, the presented classes may not be representative for other countries. In addition to that these quality classes are used especially in the context of construction work or product packaging and other areas of application as firewood, paper wood or pulp wood are not covered. The quality classes are sorted in descending order based on the market value from AC for best quality, over BR for logs with few visual quality issues, to CX for logs of minor quality with cracks or irregular shape, to the final class K with lowest quality due to beetle infestation.

For training and evaluation of the neural network a data set of *440* pile images was created with a *Samsung SM-P600 tablet* with a resolution between 640×480 pixels and 4032×3024 pixels. Most of the images were taken in front of the piles or with an offset of up to 30° according to the field of view. This dataset is distributed according to the seasons as shown in Table 1 and was manually labelled with bounding boxes. For 43 images of the data set a timestamp was not available and for this the season is not known. The used pile images contain in total *18521* and in average *42* individual logs. While the minimum amount of logs is *1*, the maximal amount is *395*. In addition to the seasonal distribution, the data set also contains logs of different types of trees and qualities. These classes are only available for a subset of the total pile data set and some logs are only classified according to the type of tree but not to its quality, which is also represented by the distribution shown in Table 2.

Table 1. The distribution of the data set according to the seasons.

Spring	Summer	Autumn	Winter	Unknown
42	125	97	133	43

Table 2. The distribution of types of trees and the associated quality of the logs in the data set.

	# Logs	AC	BR	CX	K
Spruce	2243	1464	214	13	543
Fir	15	11	0	4	0
Douglas Fir	346	270	55	2	18
Jaw	75	74	1	0	0
Larch	1135	886	76	35	52

2.1. Pre-Processing

As a first pre-processing step a sliding window approach is applied to divide the rectangular RGB input images with a 8-bit colour range into multiple square representations. This allows to uniformly avoid padding layers in the used models, if square inputs are required, and for this to increase the training performance. Afterwards the images are resampled using the Lanczos resampling algorithm as described by Fadnavis (2014) to a size of 416×416 pixels for localisation.

2.2. Data Labelling Tool

For the definition of the ground truth, a tool is needed that allows to mark the shapes, i.e. the log shapes, in the training images since we use supervised learning. In our application we define circular shapes that define the logs by centre and diameter that are transformed to bounding boxes as input for the neural net. Although there exist several data labelling tools we decided to implement our own labelling tool because it is integrated in the web back-end application that manages the wood trading process, which allows a continuous improvement of the training data set by labelling the woodpiles as they occur in the day-to-day business.

Based on the three steps in our general approach the web based labelling tool supports to add the following metadata for wooden pile images:

- for the object detection: position and radius per log as foundation for the bounding boxes
- for the segmentation: a 1-bit image mask per log

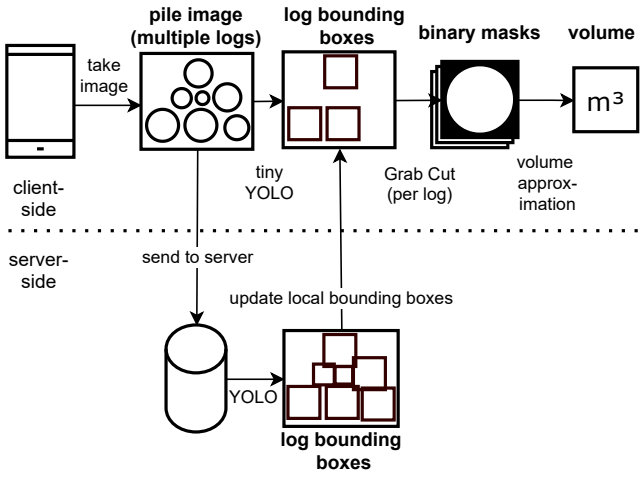

Figure 1. Process overview based on a client and a server side architecture using a tiny YOLO model for a first local user preview and fully connected YOLO on the server-side for a more accurate result.

that defines the exact and possibly irregular log shape

- for wood classification: wood species and wood quality per log in the defined range of classes

This data is stored as metadata to the wood pile and can then be exported for training. As the detection improves, the results can already be used as input for the editor, so that manual changes are only necessary for correcting the detection results.

3. Methodology

The suggested process is designed for foresters and forest owners, who are interested in the volume based value of wooden piles. Since neither a network connection nor a mobile device with high computing power can be assumed in a forest, the process is based on a separated client-server architecture. For this reason two pipelines are considered as shown in Figure 1, with one local pipeline for a first assessment based on a tiny YOLO model that is executed e.g. on a smartphone, and a pipeline on an external server, that uses a fully connected YOLO model for a more accurate result. The idea of the local pipeline is to have a very slim and fast method to get a first estimation of the number of logs in a pile. To do so the user takes an image of the wooden pile, which is distorted according to the camera model and additional sensor data as tilt and orientation. After pre-processing the image, it is used as input of the YOLO models to localize individual logs in the image. The local tiny YOLO results are replaced by the more accurate results of the server side YOLO model as soon as possible depending on the network connectivity. Based on the localization the individual logs are segmented to approximate the pile's volume.

3.1. Data Preparation

In terms of the data preparation the input data set is randomly divided into a training and test data set with a ratio of 50% to 50%. The actual separation process has to be considered from different perspectives in context of the localisation and for this the separation of logs of the same pile has to be avoided, to avert a bias of the model in terms of e.g. light or background conditions, but also according to differences in the area of the cut surface because of snow, mud or shadows. Due to that, the split into a train and a test data set is done in a first step based on pile level and secondly on log level, using the associated piles.

3.2. Data Augmentation

The data augmentation is used to increase the number and the diversity of the training data set. For this task different augmentation methods are randomly applied on the input. For example classic image processing methods as flipping, cropping, translating, rotating or adaptions of the contrast, brightness and saturation are executed. In addition to that, also more complex methods are applied for individual logs adding augmented snow or shadows. This is done, because snow- or partially shadow-covered logs exist but are underrepresented in the original data set. In terms of the snow augmentation the edges of the original log as shown in Figure 2a is covered with a randomly selected snow texture using a randomly generated mask. Such a mask is created based on a periodic oscillation along the log's circumference with n extreme points, that are randomly moved towards or away from the centre of the trunk. In the subsequent step Gaussian blur is applied and the mask is restricted to the log's shape utilizing the approximate from the labelling tool, see Figure 2.

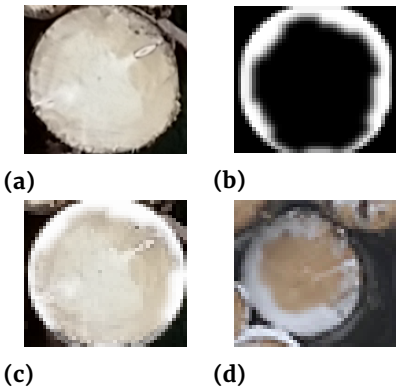

Figure 2. A clipping of a (a) log image is covered with a snow texture using a (b) randomly generated mask and results in an augmented (c) log with snow covered edges. The last image (d) shows a real world snow covered log for comparison.

Figure 2b shows a generated mask, and Figure 2c its application with a snow texture on the trunk of Figure 2a. For comparison Figure 2d shows a real snow covered log.

In addition to modelling the snow, shadow augmentation methods are applied to the input images. For this, two types of shadows are differentiated: (I) drop shadows and (II) splatter shadows. In the context of real world images the first variant may be the result of other more prominent logs in the pile, while the second variant can happen due to branches in the forest, where the pile was photographed. In terms of the augmentation process, drop shadows are created based on a randomly chosen intensity, angle and shadow length. For this method the brightness of the input image is decreased along a gradient that is defined by the given parameters from the log's centre point to the image's border along the given angle. In addition to the mentioned parameters, a binary mask allows to restrict the augmentation to a certain area, see Figure 3.

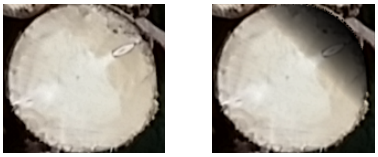

Figure 3. A clipping of a (a) log image is manipulated using the drop shadow augmentation method as shown in (b).

The second shadow augmentation method adds a splattered shadow to the image. For this method n random seed points are added to an image mask with the size of the input image. Based on this seed points, areas are grown in random directions. This is done in a region growing like way as described by Adams and Bischof (1994), in which neighbour pixels are randomly added to the area until a certain amount of pixels has been included. In the next step the so created binary mask is blurred in two ways, using the Gaussian Blur algorithm and an additional motion blur algorithm. The so created mask is applied to the image as shown in Figure 4. This is to model the smoothed light distribution due to leaf work and to model the motion of leaves in the wind. Based on the created mask the same approach can be used for applying textures and like so to add mud as shown in Figure 4c or snow as shown in Figure 4d to the trunk. As comparison to the augmented variants you can see real world examples of mud on a log in Figure 4d and for snow in Figure 4f.

3.3. Multi-Resolution Localization

A YOLO model is used as first step of the pipeline independent of the place of execution (client- or server-side). This neural network is applied on overlapping

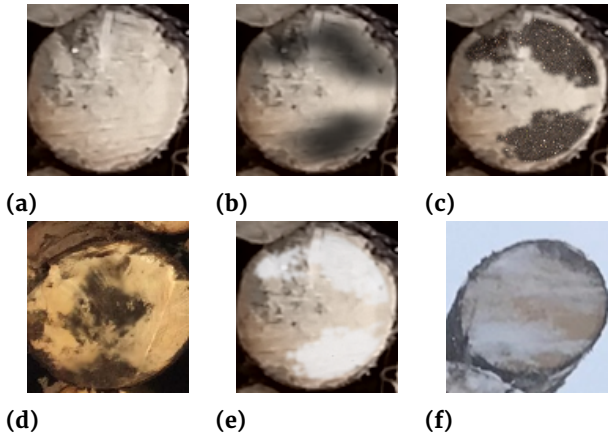

Figure 4. A clipping of (a) a log image is manipulated using the splattered shadow approach as shown in (b), which is in turn used as alpha mask for applying a (c) mud texture, that can be compared with (d) a real world mud image. Next to the mud augmentation it also allows to use e.g. (e) a snow texture. For this (f) shows a real world log with snow on the surface for comparison.

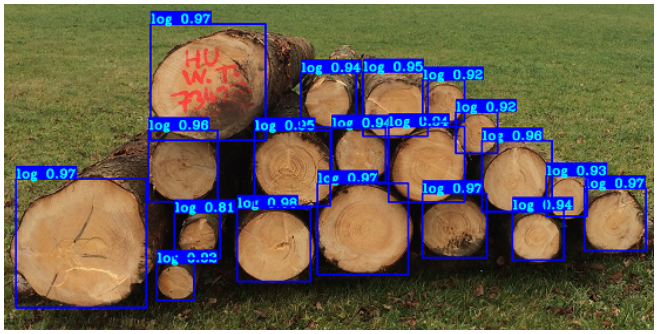

Figure 5. The YOLO model is used to detect bounding boxes around individual logs of a wooden pile giving also the detection confidence.

tiles of the input pile image. The tiles are created using a sliding window approach and allow to detect smaller logs as well as big ones due to the ratio reduction of the logs' size in reference to the size of individual observed regions of interest. YOLO is used to locate the logs in these clippings and results in bounding boxes describing the positions and the confidence of the detected logs, as shown in Figure 5. The only difference in this step between the two proposed pipelines is that a tiny version is used on the local side, while on the server-side a classic YOLO model is used as described by Wang et al. (2020). This leads to the situation that a first user preview is even possible on a smartphone, for which accuracy losses are accepted in favour of speed and highly decreased memory consumption.

3.4. Tiling

It is a common problem of the YOLO object detection model that there can occur difficulties with detecting small objects whereas the limit is not given in absolute pixels since detection algorithm downsizes the

image to 416 × 416 pixels relative to the image size. One solution for that problem is to reduce the number of layers and to increase the number of strides in the model, which can reduce the models accuracy. Alternatively, the main image can be separated into overlapping tiles on which we apply the YOLO detection so that the smallest logs are above the critical size, finally the results given by the individual tiles are merged into the main image. The count of tiles by rows and columns and the overlapping is configurable since settings can vary strongly between the different trained nets due to different lower and upper bounds of detectable object sizes.

Practical tests with darknet YOLOv4-tiny gave good results with 3 by 2 tiles and overlapping that corresponds to the biggest logs to be detected.

Furthermore it is planned to evaluate multi-resolution approaches with increasing subdivisions per level (1×1, 2×2, 3×3, 4×4) images which results in a higher robustness of the algorithm. On the other hand this comes with additional computational cost since it would result in 30 detection rounds and has to be evaluated on the handheld device. On the server side we have less restrictions concerning computation resources.

4. Implementation

The implementation of the presented augmentation approach was done using Python 3.7 and OpenCV 4.5.1.48. For the training of the neural networks the Darknet framework by Bochkovskiy (2021) and TensorFlow 2.3.1 were used. These machine learning frameworks were utilized to create the (tiny) scaled YOLOv4 models. The models are trained on an environment using an Intel Core i9-10900K and a Gigabyte GeForce RTX 3070 with images of 416 × 416 pixels for the localization.

5. Results

The presented approach is evaluated using both models, the tiny YOLO model used in the local pipeline, as well as the full YOLO from the serverside, based on 10 sample images (shown in Figure 6) containing 985 logs in total with a log area of 5161027 pixels. To our knowledge, there is no comparable dataset publicly available. For comparison, the sample images are analysed using Haar Cascades as presented by Auersperg-Castell (2018), too. The results in Table 3 show that the Tiny YOLO network is able to detect 844 logs with 27 false positives according to the ground truth. This leads to the situation that 82% of the individual logs and 72% of the log area pixels are detected correctly. Additionally, the fully connected YOLO results in a detection rate of 90% according to the log area and is able to detect 949 logs with 22 false positives, so 94% of the logs are detected correctly. The variance of the correctly detected bounding boxes according to the ground truth bounding boxes results in 0.02 for the tiny YOLO model and 0.016 for the fully connected model.

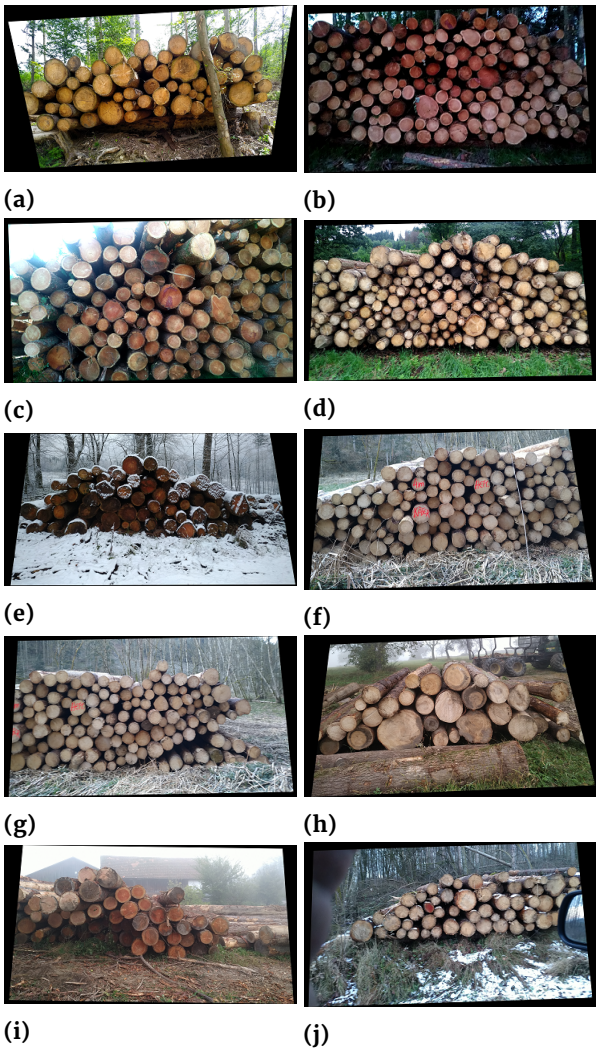

Figure 6. The wooden pile data set used for the evaluation of the presented approach with 985 logs in total.

6. Discussion

As shown in the results we were able to tackle the problem of detecting individual logs in wooden pile images. The results also show that we are able to localize small logs due to the usage of the proposed tiling process and for this can successfully address the first research question "Can YOLO be used in a multi-resolution approach to allow for localization of small wood logs?". Due to the separation into a client-server architecture and the utilization of different YOLO models we are able to get on the one hand first previews in real-time, but on the other hand also more accurate results using the server-side model and can for this also tackle the second research question "Can consumer smartphones be

Table 3. Comparison based on the evaluation data set with 985 logs shown in Figure 6 using a classic Haar Cascade approach and the presented YOLO based methods with an accuracy of 82,9% detected logs and 72% according to the detected log area for the tiny YOLO model and 94,1% respectively 90% for the fully connected YOLO.

	Detected Logs	False Positives	Missing Logs	Correct Logs	Area Variance	Detected Area (in pixels)	Area Ratio
Haar Cascade	712	23	296	69,9%	0.03288	3771790	54%
Tiny YOLO	844	27	168	82,9%	0.02085	4969932	72%
Full YOLO	949	22	58	94,1%	0.01699	4771724	90%

utilized to achieve preview results in real-time that are refined on a server for high-quality results?". Finally, the results also show that we are able to compensate seasonal wood characteristics using specialized data augmentation techniques to increase the samples in the training data set for a more robust YOLO model. This in turn leads to a positive answer to the third research question "Can lack in seasonal wood log characteristics (snow, mud,...) and weather conditions for the acquired images be compensated by advanced data augmentation strategies?".

7. Summary and Outlook

As shown in this paper, state of the art computer vision algorithms for localization can be seamlessly integrated to facilitate a scale-invariant multi process analysis approach. Thereby, the utilization of heterogeneous machine learning frameworks and programming environments is conquered. The separation of the problem into localization with subsequent segmentation / classification allows to boost the overall quality of results and further facilitates a client / server infrastructure where preview results can be provided on common smartphones in real-time while further analysis and higher accuracy are asynchronously performed on a server.

In future, the focus of research and development will be laid onto the self-adapting nature of the algorithms. With the presented tools for data labelling, the results of the DL algorithms can be visually inspected and post-processed in a semi-supervised way thus allowing to incrementally enrich the training data sets of the models getting sequentially re-trained.

8. Funding

Many thanks to the local government of Upper Austria for facilitating this research initiative in the course of *easy2innovate* funding program.

9. Acknowledgements

Special thanks to *Ulrich Hainberger* and *Luis Hainberger* from the Ulrich Hainberger e.U. forestry company for support in the cooperation, for providing thousands of precious heterogeneous test images and for invaluable input and great discussion.

References

Adams, R. and Bischof, L. (1994). Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6):641–647.

Auersperg-Castell, P. (2018). Photooptische holzpoltervermessung mittels haar-kaskaden. Bachelor's thesis, University of Passau, Germany.

Ballard, D. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122.

Bochkovskiy, A. (2021). Alexeyab/darknet.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166.

Cengil, E. and Çinar, A. (2017). Comparison of hog (histogram of oriented gradients) and haar cascade algorithms with a convolutional neural network based face detection approach. *International Journal of Advance Research, Ideas and Innovations in Technology*, 3:244–255.

Chen, W., Chen, S., Guo, H., and Ni, X. (2020). Welding flame detection based on color recognition and progressive probabilistic hough transform. *Concurrency and Computation: Practice and Experience*, 32(19):e5815.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893.

Daugman, J. (2006). Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, 94(11):1927–1935.

Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey.

Fadnavis, S. (2014). Image interpolation techniques in digital image processing: an overview. *International Journal of Engineering Research and Applications*, 4(10):70–73.

Flämig, H. (2016). *Autonomous Vehicles and Autonomous Driving in Freight Transport*, pages 365–385. Springer Berlin Heidelberg, Berlin, Heidel-

berg.
He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask R-CNN. *CoRR*, abs/1703.06870.
Mahmoud, A., Mohamed, S., El-Khoribi, R., and Abdelsalam, H. (2020). Object detection using adaptive mask rcnn in optical remote sensing images. *Int. J. Intell. Eng. Syst*, 13:65–76.
Murata, K., Ito, E., and Fujimoto, T. (2018). A proposal for "infinite scale" ruler application that provides analog-like "sensory reality". In *2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 925–928.
Ojala, T., Pietikäinen, M., and Harwood, D. (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *Proceedings of 12th International Conference on Pattern Recognition*, 1:582–585 vol.1.
Paredes-Astudillo, Y. A., Jimenez, J.-F., Zambrano-Rey, G., and Trentesaux, D. (2020). Human-machine cooperation for the distributed control of a hybrid control architecture. In Borangiu, T., Trentesaux, D., Leitão, P., Giret Boggino, A., and Botti, V., editors, *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*, pages 98–110, Cham. Springer International Publishing.
Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.
Reithmeier, L., Krauss, O., and Zwettler, G. A. (2021). Transfer learning and hyperparameter optimization for instance segmentation with rgb-d images in reflective elevator environments. In *Proc. of the WSCG2021*.
Reitinger, B., Werlberger, P., Bornik, A., Beichel, R., and Schmalstieg, D. (2005). Spatial measurements for medical augmented reality. In *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)*, pages 208–209. IEEE.
Sakinis, T., Milletari, F., Roth, H., Korfiatis, P., Kostandy, P., Philbrick, K., Akkus, Z., Xu, Z., Xu, D., and Erickson, B. J. (2019). Interactive segmentation of medical images through fully convolutional neural networks.
Schmucker, M., Igel, C., and Haag, M. (2019). Evaluation of depth cameras for use as an augmented reality emergency ruler. In *dHealth*, pages 17–24.
Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:I–511.
Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2020). Scaled-yolov4: Scaling cross stage partial network. *arXiv preprint arXiv:2011.08036*.
Zwettler, G., Backfrieder, W., Karwoski, R., and Holmes, D. (2021). Generic user-guided interaction paradigm for precise post-slice-wise processing of tomographic deep learning segmentations utilizing graph cut and graph segmentation. In *VISIGRAPP*.
Zwettler, G., Holmes III, D., and Backfrieder, W. (2020). Strategies for training deep learning models in medical domains with small reference datasets. *Journal of WSCG*, 28(1-2):37–46.