DCNN-based Screw Classification in Automated Disassembly Processes

Erenus Yildiz[a] and Florentin Wörgötter[b]
III. Physics Institute, Georg-August University of Göttingen, Germany

Keywords: Screw Classification, Automation, Disassembly, Recycling, E-Waste.

Abstract: E-waste recycling is thriving yet there are many challenges waiting to be addressed until high-degree, device-independent automation is possible. One of these challenges is to have automated procedures for screw classification. Here we specifically address the problem of classification of the screw heads and implement a universal, generalizable, and extendable screw classifier which can be deployed in automated disassembly routines. We selected the best performing state-of-the-art classifiers and compared their performance to that of our architecture, which combines a Hough transform with the top-performing state-of-the-art deep convolutional neural network proven by our experiments. We show that our classifier outperforms currently existing methods by achieving 97% accuracy while maintaining a high speed of computation. Data set and code of this study are made public.

1 INTRODUCTION

A very significant and challenging quest of the last three decades has been E-Waste recycling. Aside from being environmentally friendly, the challenge is also economically rewarding and scientifically interesting. Every year the life cycles of electronic products are slightly decreasing (Solomon et al., 2000), leading to massive amounts of valuable raw materials if they are recycled. On the other hand, the increased pace of production in electronic device industry inevitably motivates us to look for new technologies and paradigms to bring a possible solution to the automated recycling challenge. According to recent statistics, only 20% of the E-Waste are recycled (Kahhat et al., 2008). There are many reasons behind this ratio, however, here we focus on the aspects where the AI community could contribute to a possible solution. Currently most of the recycling paradigms follow a destroy-then-melt strategy, without much intelligence involved. Intelligent automation could not only increase the percentage of recycling, it could also let better recycling paradigms evolve.

Economical reasons are definitely not the only reasons to push forward for automated recycling. According to a recent study (Jahanian et al., 2019), melting 1 million phones could potentially recover 16,000kg of copper, 350kg of silver, 34kg of gold and 15kg of Palladium. However, this type of solution usually causes environmental and health hazards in and around disassembly plants. Additionally, they ignore the possibility of finer recycling due to the fact that they prefer low-cost rather than higher revenue. Then, ultimately, there is a necessity of building intelligent solutions to automate detection and categorization of objects that are pivotal to the disassembly routines such as screws, so that the autonomous robotic manipulation tasks (i.e. unscrewing) are carried out successfully.

It is indeed an interesting problem to look into since it is challenging to address detection and classification of screw heads due to their different designs. Consider a computer hard drive containing screws of different types and sizes, posing a great challenge of finding features with high accuracy to automate the unscrewing action, which is frequently carried out by humans. Screws have fixed poses (thus limited viewpoints) and only their heads are visible to the capturing sensor, making it a problem of analysing the features available on the head surface to figure out the type and size information. These features are sometimes really challenging to catch due to the small sizes of the screw heads (i.e. Torx7 vs. Torx8). Additionally, they are not only small, but also shiny objects and, thus, reflecting much of the light. All in all, it becomes really difficult to capture all the necessary information with an effective precision without using a several-thousand-dollar high-end line-laser.

[a] https://orcid.org/0000-0002-3601-7328
[b] https://orcid.org/0000-0001-8206-9738

Yildiz, E. and Wörgötter, F.
DCNN-based Screw Classification in Automated Disassembly Processes.
DOI: 10.5220/0009979900610068
In *Proceedings of the International Conference on Robotics, Computer Vision and Intelligent Systems (ROBOVIS 2020)*, pages 61-68
ISBN: 978-989-758-479-4

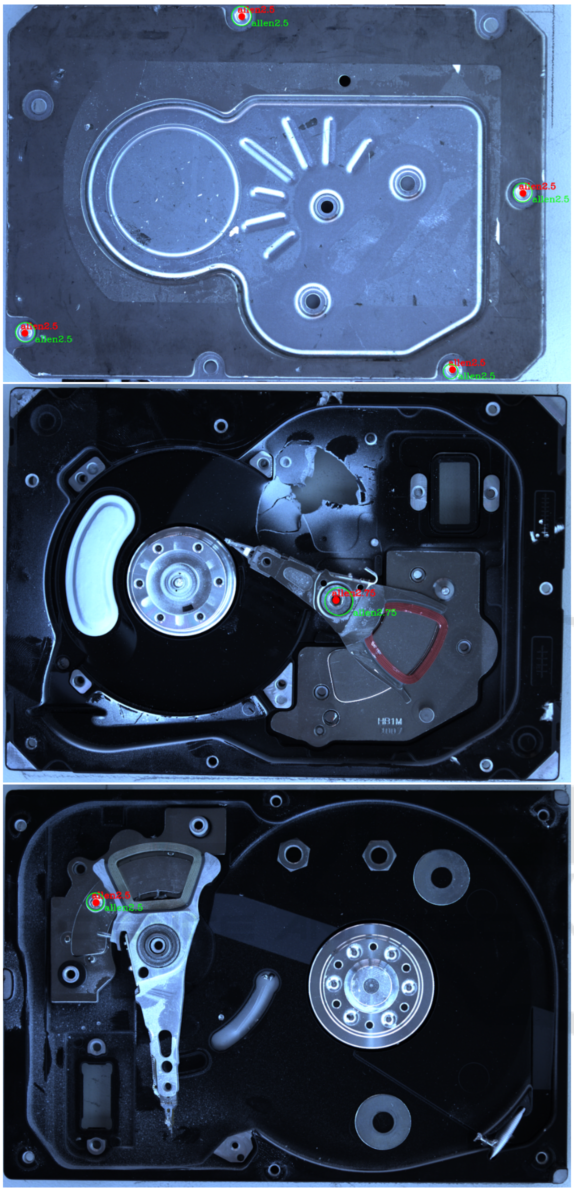

Figure 1: We present a universal, extendable, RGB based screw head classifier scheme which is engineered for automated disassembly tasks. Our scheme uses a Hough transform and two state-of-the-art deep convolutional neural networks. Green circles represent predicted screw type, red dot and font represent the ground truth.

In this paper, a visual screw head classification scheme based on a combination of deep learning methods empowered with classical computer vision methods is proposed. The proposed scheme can be seen as an extension of a previous scheme developed by us (Yildiz and Wörgötter, 2019) to detect screws. We take the work even further now by making it able to classify the screw type and size. We use the well known Hough transform to first generate screw-candidates (screws and circular artefacts), which are then filtered out by our previous screw detector model, yielding only screws. Afterward, we feed the screws into our DCNN powered classifier, chosen via our experimental evaluation of the top state-of-the-art models. The scheme can account for any type of screw size and type as long as the user collects enough data in the offline mode for training. As it was in our previous work, we keep the feature of data collection. Contrary to state-of-the-art techniques, which require users to find datasets from search engines for their specific requirements (viewpoint, height, etc.), we let users create their own datasets, given the device and camera. This mitigates the problem of finding specific datasets for screw types and sizes, and ensures high accuracy for the network.

Therefore, the most significant contribution of this paper is the fusion of features acquired from classical computer vision methods and the deep learning model that trains on the data collected by the user in a semi-automated way. To our knowledge, we are the first to utilise this kind of fusion to classify 12 different screw head types.

We examine the performance of our proposed scheme in different situations and the extent of generalization for effective automation and robotics usage in disassembly. For our experiments, we have collected approximately 35000 training images of different screw head types and sizes. Additionally, 50 sample scenes of various disassembly stages of computer hard drives were collected to test the pipeline.

2 RELATED WORK

There have been several studies researched under the topic of automated disassembly (Weigl-Seitz et al., 2006; Dröder et al., 2014; Wegener et al., 2015) which yielded some schemes (Elsayed et al., 2012; Pomares et al., 2004; Bdiwi et al., 2016; Büker et al., 2001) for automating certain processes. However, these schemes do not generalize and pose as universal solutions for the problem of classification of screw heads. Basically, the problem of screw head classification is not even addressed yet by the community. There are some works that involve template matching methods to find the screws on metal ceiling structures for dismantling of certain objects (Ukida, 2007). However, template matching operates on a fixed template that looks for a pixel-level match with the target, making it an extremely undesirable paradigm for cases where lightening or the object's color changes. It is therefore impossible to account for all screw heads of all colors, sizes, lightening conditions by using template-matching based methods, making it an impossible scheme for automated disassembly. An interesting attempt came with the goal of detecting M5 bolts on battery joints in electric vehicle battery disassembly routines (Wegener et al., 2015). Using a Haar-type cascade classifier trained on cropped images of M5 bolts, adding false positives detected from the classifier under negatives, the authors were only able to achieve 50% detection accuracy for only one type of bolts. Such low accuracy does not constitute a viable solution for a disassembly routine, thus making the method impractical for industrial use.

The last work we would like to mention also focused on autonomous disassembly of electric vehicle motors (Bdiwi et al., 2016). In this work, screws found on electric vehicle motors were detected using an RGB-D sensor (Kinect) (Zhang, 2012). Although the proposed algorithm is scale, rotation, and translation invariant, it heavily relies on traditional computer vision methods such as Harris corner detection and HSV image analysis. It is a well known fact that these methods are easily affected by the lighting conditions and do not generalize. Another shortcoming is the fact that they require a depth image from the RGB-D sensor to remove false positives such as holes, which adds computational load, yet again for merely detecting a screw, not classifying its type and size.

Thus, it seems that there is still a substantial lack in generalizable, device and environment-independent methods to detect and classify screw heads, which can be used in automated disassembly processes.

3 METHOD

In this section we explain each block in our pipeline. However, before doing that, we would like to inform the reader about the setup our scheme requires. We propose a setup in which the camera faces the device's surface perpendicularly. The distance between the device and the camera was 75 cm, however, depending on the size of the device, this distance may change. Since we worked with computer hard drive screws, 75 cm was a suitable height.

We keep the pipeline we inherit from our previous work(Yildiz and Wörgötter, 2019), and add a classification block to it, as illustrated in Fig. 2. Our previous work enables the user to collect training data by cropping circular candidates from the scene. The cropped circular candidates are then to be divided into their respective classes (artifact, Torx8, Ph2, Slotted6.5, Allen2.75, etc.) by a human.

Afterward, first the detector model can be trained to classify screws from artefacts (circular non-screws structures), as we explained in our previous work (Yildiz and Wörgötter, 2019). Having deployed a model that can differentiate screws from artefacts, now we can train and deploy our new classifier, that can classify 12 different types of screw heads. Our scheme then marks and returns the type/size information and locations of the screws seen in the image.

3.1 Preprocessing

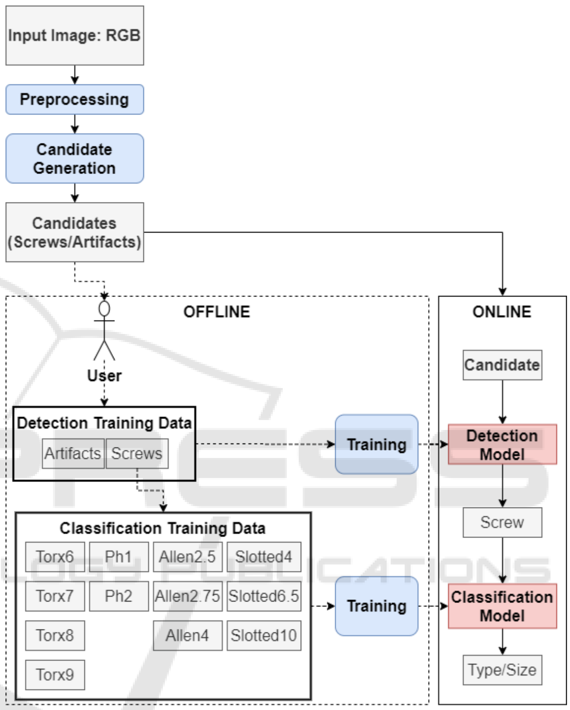

Figure 2: The pipeline in our scheme is composed of online and offline stages. Online stage employs models to differentiate screws from artefacts, as well as to classify the screws.

The preprocessing step is directly taken from our previous work (Yildiz and Wörgötter, 2019), where classical computer vision is first used to crop the image to only the region where the device is visible, and grayscale it.

3.2 Candidate Generation

E-Waste is a vast category of devices with different structures and materials. A possible screw head classifier should therefore account for all or most of the screws encountered in the disassembly domain. In order to come up with a scheme that has reasonable levels of generalisation ability. We analyzed different types of screws found in the domain of E-Waste. To make sure that our method will cover the most encountered and conventionally used screw types found in this domain, we have consulted experts from the disassembly plant in cooperation with our university and agreed on 12 types of screw heads such as different sizes of Torx, Philips, Slotted and Allen heads. We assessed various electronic devices, which can be found in huge numbers nowadays in E-Waste, such as computer hard drives, DVD players, gaming consoles and many more. As anticipated, we concluded that almost all screws in this domain are circular, which is the natural geometry of these objects and represents the central feature to be utilised to detect a screw object. Fig. 3 illustrates samples of screw types/sizes classified in our dataset. It must be underlined that there are also non-circular screws manufactured, however, those are few and we found no such screws in the devices of interest in the disassembly plant we cooperate with. We therefore based our method on first finding circular structures in the images. Clearly, not every circular structure is a screw, for example stickers, holes, transistors, etc. exist, which are also circular, but not screws. Nevertheless, circular structures provide us with priors for screws and the first step of our method is to collect those screw-candidates.

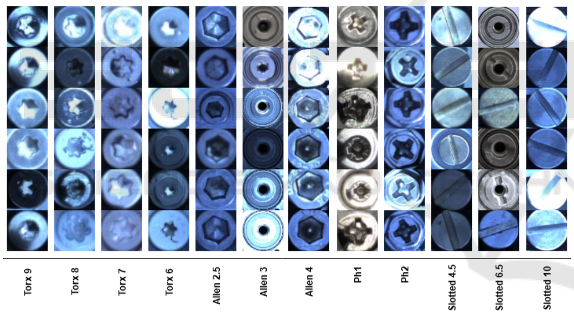

Figure 3: Screw types encountered during the disassembly of various electronic devices found in E-Waste. Last row depicts artefacts that count as another type in classification.

We use the base candidate generator from our previous work (Yildiz and Wörgötter, 2019), in order to collect candidates. We run our program in offline mode and rely on the *Hough Transform* for candidate detection. This is a standard computer vision method for circle detection (Duda and Hart, 1971) and shall not be explained here. Differing from the standard Hough Transform here we use a version, relying on the so called *Hough Gradient* (of the OpenCV library (Bradski and Kaehler, 2008)). This version uses the gradient information of the edges that form the circle. We refer the reader to the handbook published by the creators of the aforementioned library for further implementation details on the algorithm of the Hough Gradient. It should be noted that after collecting our candidates, we then switch back to RGB from grayscale, since our classifiers operate far better in colored images than in grayscaled ones.

3.3 Training the Classifiers

As mentioned before, the user manually separates screw types and artifacts by which a classifier can be trained using these positive and negative examples as training data. In Fig. 3 types of screw heads and artifacts taken from various devices found in E-Waste are shown. In general, these screws are found in other device-classes and, thus, the resulting training set can be transferred also to other devices. In that case, however, one has to increase the number of samples to account for more types of screws.

We have investigated state-of-the-art classifiers found in the literature and we picked the three top-performing ones for comparison at the end. These networks, to our experience, were performing tolerably good given a not so large dataset for a specific device-class (hard drives of any size). Finally, we decided to evaluate EfficientNets (Tan and Le, 2019), ResNets (He et al., 2016), DenseNets (Huang et al., 2017), scoring top accuracies on the ImageNet (Deng et al., 2009). Additionally, EfficientNets have been used in the latest works (Xie et al., 2019) in pursuit of improving ImageNet classification, by using a new self-traning method called Noisy Student Training. Inspired by this effort, we chose EfficientNetB2.

Our strategy to evaluate the networks is described as follows. We go by the standard procedure for transfer learning: cutting the pre-trained model on the last convolutional layer and adding a new sequence of linear layers called the head. We use this head architecture for all models we explore. In the first 10 epochs, we train only the added final layers of the model by freezing all convolutional layers, not allowing any updates to their weights. Afterward, we unfreeze all layers and train the entire network. We find it useful to use differential learning rates at this stage. It is not desired to change the early layers of the models as much as the later ones, therefore lower learning rates are used in the first layers and higher ones in the end. Using the Adam optimizer (Kingma and Ba, 2014) with the learning rate of 1×10^{-5}. Figure 4 illustrates the model architecture we use. Here, the term ”Block” is a higher abstraction used for group of layers.

To further reduce overfitting and to come up with a model that can generalize, we applied an additional data augmentation step. There are several data augmentation operations we applied to introduce more variety in the data such as rotation, brightness and contrast.

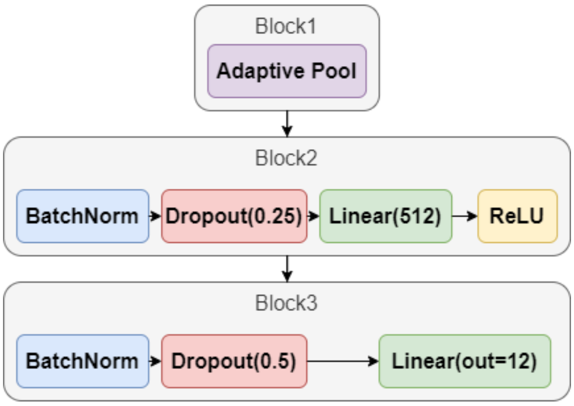

Figure 4: Head architecture of the model.

4 EXPERIMENTAL EVALUATION

We conducted several experiments on the test data we collected. Out of the top three state-of-the-art classifiers, we picked the best performing one, namely EfficientNetB2 -one of the best performing models given Noisy Student weights- and used it in the pipeline seen in Fig. 2. Below we also provide details of the experimental evaluation and present our justification for our decision of picking the one we use in our pipeline.

4.1 Experimental Environment

For the evaluation of the screw classifiers, we collected a dataset of over 20000 samples and split it into training and validation sets with the ratio of 2:1. We use a computer with Intel Core i7-4770 CPU @ 3.40GHz, 16GB of RAM with GeForce GTX Titan X graphics card to train the classifiers. For evaluation of the performance of our entire pipeline, we collected approximately 50 hard drive images containing over 500 screw-like elements as the test set the model has never seen before.

4.2 Experimental Metrics

Our pipeline is composed of two main blocks, namely the Hough circle detector and our classifier, EfficientNetB2. It is required to assess the detection as well as the classification abilities of the entire pipeline. To this end, we went by the following strategy: We first annotated the test images, each having only one hard drive with top-down view. These images contain drives with or without screws, by which the Hough circle finder could be assessed. We annotated these scene images by marking screws with squares, which would form our ground truth for assessing the Hough circle finder's accuracy. We went by the standard VOC evaluation (Everingham et al., 2010) and we found our Hough circle detector to work with 0.783 mean IoU with the optimal parameters found for our setup. IoU here refers to what amount of screw region is correctly detected by the Hough circle finder. If the detected region for a screw is below 70% it's bound to result in bad prediction for both detection and classification. It must be also noted that our pipeline is limited by the accuracy of Hough, since if the circle finder cannot catch the circle, then the classifier's accuracy does not matter at all. The accuracy of Hough circle detection can vary depending on the parameters of the function such as min/max radius, min/max threshold. Inevitably, final accuracy of our pipeline can be calculated as follows:

$$Acc\_Pipeline = Acc\_CircleDetector * Acc\_Classifier$$

For the evaluation of the classifier, we consider the standard metrics of loss, accuracy, min_accuracy and f1_score.

Table 1: Accuracy of the state-of-the-art models with huge variation of hyperparameters. Highlighted ones are the top three performing ones.

Model	Grayscale	Size	Loss	Acc.	Min. Acc.	F1	Transfer Learning
EfficientNetB2A	**No**	**256**	**0.1187**	**0.968**	**0.79**	**0.97**	**Noisy Student**
EfficientNetB2A	No	64	0.2144	0.936	0.78	0.93	ImageNet
EfficientNetB2A	No	128	0.1871	0.951	0.85	0.95	ImageNet
EfficientNetB2A	Yes	128	0.2199	0.948	0.67	0.94	ImageNet
EfficientNetB3A	Yes	64	0.2072	0.937	0.75	0.93	ImageNet
EfficientNetB3A	No	64	0.2051	0.939	0.74	0.94	ImageNet
DenseNet121	**No**	**128**	**0.1415**	**0.961**	**0.81**	**0.96**	**ImageNet**
DenseNet121	Yes	128	0.1489	0.957	0.74	0.95	ImageNet
DenseNet121	No	64	0.1896	0.937	0.72	0.93	ImageNet
DenseNet121	No	64	0.2306	0.934	0.71	0.93	ImageNet
DenseNet201	No	256	0.1170	0.966	0.79	0.96	ImageNet
ResNet34	**No**	**128**	**0.1538**	**0.955**	**0.80**	**0.95**	**ImageNet**
ResNet34	Yes	128	0.2026	0.951	0.69	0.95	ImageNet
ResNet50v2	No	256	0.1732	0.942	0.73	0.94	ImageNet

4.3 Experimental Results

We summarize the experimental results with regards to performance of each classifier against the validation set in Table 1. Note that we are selecting what model to use in this part of our experiments. Using the test dataset would cause overfitting of the hyperparameters.

From the collected results in Table 1, one can conclude the following: All of the investigated models achieve very high accuracy - over 90% on the testing data, with the model EfficientNetB2 scoring the highest min. accuracy of 85% among single models. Additionally, we emphasize that augmentation strategy plays a pivotal role in the classifier accuracy. Especially for circular objects, rotation guarantees that the traning data accounts for screws that are rotated for each angle. Using the Albumentations (Buslaev et al., 2020) library applied a rotation of 360 degrees, horizontal and vertical flips, as well as brightness and contrast changes.

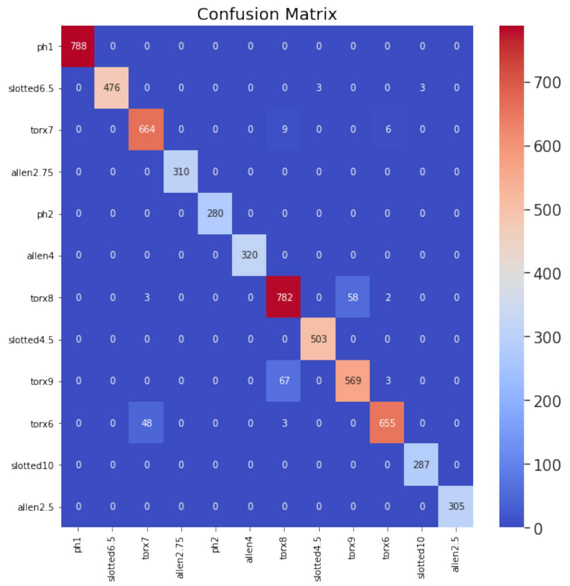

Figure 5: Confusion matrix acquired through our experiments.

We then employ the final model in our pipeline and evaluate it on our test dataset. Fig. 6 shows the precision-recall curve of our classifier, whereas Fig. 5 illustrates the confusion matrix. Our pipeline achieves an AP of 0.757. The reason why our classifier accuracy is higher than our AP is due to the Hough circle finder, which succeeds by approximately 75% of the time, limiting our pipeline's overall AP. This can be mentioned as the only limitation of our pipeline. Fig. 7 illustrates the detection and classification of the screws found on a Hitachi Deskstar 3.5" hard drive during the disassembly.

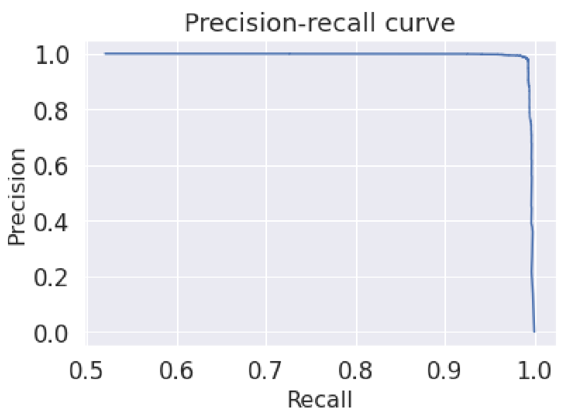

Figure 6: Precision-Recall curve for the final model for our classifier.

5 DISCUSSION AND CONCLUSIONS

In this study we tackled the fundamental problem of screw head classification in disassembly environments. The problem itself is a challenging one, since screw heads have variable sizes, types and not every electronic device has the same type of screw heads. We proposed a model, which is based on the Hough transform and two DCNNs. Our proposal was the extension of the previous work we published, where the scheme lets user to collect as much as data he/she desires from a device. Given that the user creates the ground truth classification (i.e., separates the collected screw data into their respective type and size categories), our model is able to hit very high F1 score of 97%, whereas with Hough we found the optimal parameters to hit 78% IoU for screw regions. The results had been quantified with hard drive devices of different models and sizes, which have different sizes and types of screws as documented by the experimental evaluation results of our scheme. Additionally, we acquired screws from other devices and inserted them into the scene to test the generalization ability of the pipeline, achieving promising results as well. The data set as well as the implementation are to be published to facilitate further research[1].

As future work, we note that Hough circle finder can be employed in a better way. An alternative to our current strategy with Hough circle finder would be shifting of detected circle ROI to conduct dynamic region correction in order to hit higher IoU. This is due to the fact that candidates suggested by Hough may be cut off in a way to mislead the classifier.

Note that all the misclassifications apart from the intra-class ones (e.g., Torx6/Torx8) are empirically found to be a direct result of how Hough cuts regions. In some cases the artefacts are found as Allen2.75, which, however, is a strongly valid classification and detection by the classifiers since the candidate cropped by the Hough circle resembles an Allen screw, and, thus the classifier claims so. The suggested approach therefore could provide tangible results by improving the cropping of candidates, however, it is a different research topic altogether, and, mostly lies in the area of algorithm building.

[1] https://drive.google.com/drive/folders/1VFrF7B1FXPTgJRpu8rcnxJFRRKiT2ljY?usp=sharing

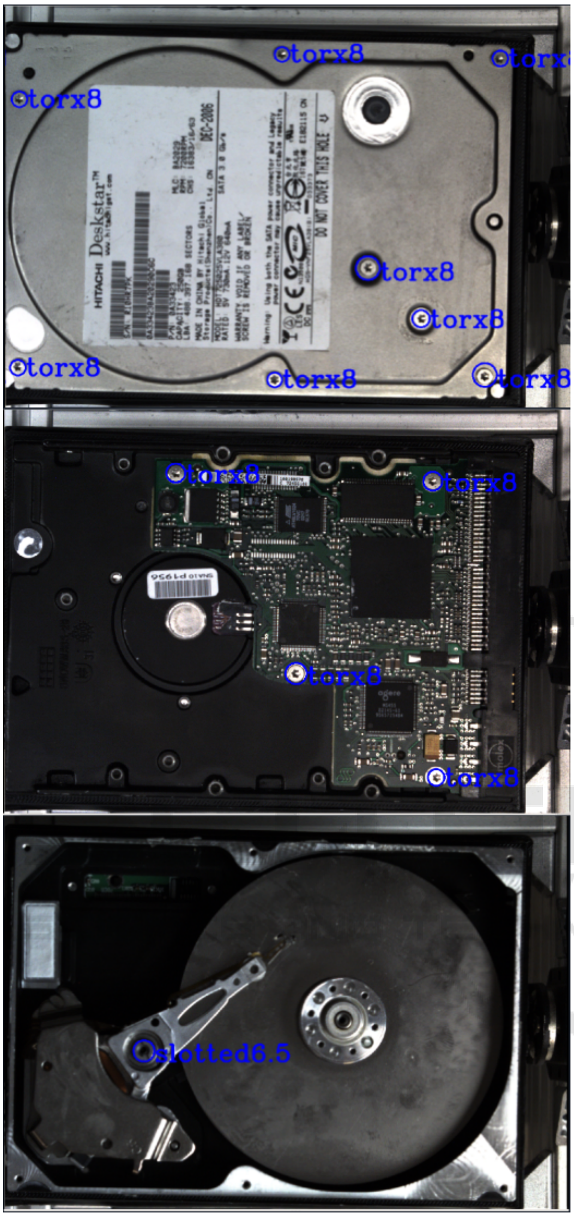

Figure 7: Classifications of detected screws during the disassembly of a hard drive. All screw types are found correctly.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation programme (H2020-ICT-2016-1, grant agreement number 731761, IMAGINE; https://imagine-h2020.eu/). We would like to also express our gratitude to the disassembly plant we cooperated with, namely Electrocycling GmbH[2], for their continuous support throughout our investigation.

REFERENCES

Bdiwi, M., Rashid, A., and Putz, M. (2016). Autonomous disassembly of electric vehicle motors based on robot cognition. *Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June(July):2500–2505.

Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.".

Büker, U., Drüe, S., Götze, N., Hartmann, G., Kalkreuter, B., Stemmer, R., and Trapp, R. (2001). Vision-based control of an autonomous disassembly station. *Robotics and Autonomous Systems*, 35(3-4):179–189.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. *Information*, 11(2):125.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Dröder, K., Raatz, A., Herrmann, C., Wegener, K., and Andrew, S. (2014). Disassembly of Electric Vehicle Batteries Using the Example of the Audi Q5 Hybrid System. *Procedia CIRP*, 23:155–160.

Duda, R. O. and Hart, P. E. (1971). Use of the hough transformation to detect lines and curves in pictures. Technical report, SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER.

Elsayed, A., Kongar, E., Gupta, S. M., and Sobh, T. (2012). A robotic-driven disassembly sequence generator for end-of-life electronic products. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 68(1):43–52.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Jahanian, A., Le, Q. H., Youcef-Toumi, K., and Tsetserukou, D. (2019). See the e-waste! training visual intelligence to see dense circuit boards for recycling. In *Proceedings of the IEEE Conference on Computer*

[2]www.electrocycling.de

Vision and Pattern Recognition Workshops, pages 0–0.

Kahhat, R., Kim, J., Xu, M., Allenby, B., Williams, E., and Zhang, P. (2008). Exploring e-waste management systems in the united states. *Resources, conservation and recycling*, 52(7):955–964.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Pomares, J., Puente, S. T., Torres, F., Candelas, F. A., and Gil, P. (2004). Virtual disassembly of products based on geometric models. *Computers in Industry*, 55(1):1–14.

Solomon, R., Sandborn, P. A., and Pecht, M. G. (2000). Electronic part life cycle concepts and obsolescence forecasting. *IEEE Transactions on Components and Packaging Technologies*, 23(4):707–717.

Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks.

Ukida, H. (2007). Visual defect inspection of rotating screw heads. In *SICE Annual Conference 2007*, pages 1478–1483. IEEE.

Wegener, K., Chen, W. H., Dietrich, F., Dröder, K., and Kara, S. (2015). Robot assisted disassembly for the recycling of electric vehicle batteries. *Procedia CIRP*, 29:716–721.

Weigl-Seitz, A., Hohm, K., Seitz, M., and Tolle, H. (2006). On strategies and solutions for automated disassembly of electronic devices. *International Journal of Advanced Manufacturing Technology*, 30(5-6):561–573.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. (2019). Self-training with noisy student improves imagenet classification.

Yildiz, E. and Wörgötter, F. (2019). Dcnn-based screw detection for automated disassembly processes. In *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 187–192. IEEE.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10.