Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira,
Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria
 Published by Research Publishing, Singapore.
doi: 10.3850/ESREL2026061419_esrel26-p26334-cd

An Application of Semi-Supervised Learning to Convolutional Neural Networks for Surface Roughness Classification

Benedikt Bubert

Chair for Artificial Intelligence in Mechanical Engineering, Munich University of Applied Science, Germany.
E-mail: benedikt.bubert@hm.edu

Marcin Hinz

Chair for Artificial Intelligence in Mechanical Engineering, Munich University of Applied Science, Germany.
E-mail: marcin.hinz@hm.edu

Knowledge about the quality of machine-produced products is crucial for manufacturers. Since the quality of certain parts cannot be determined non-destructively using conventional inspection methods, the demand for non-destructive quality control solutions is continuously rising. Due to recent advancements in deep learning, these methods can now be utilized for non-destructive quality inspection. Specifically, Convolutional Neural Networks have demonstrated exceptional performance in image classification tasks. Therefore, we use a Convolutional Neural Network to evaluate the quality of fine-ground chef knife surfaces by predicting the arithmetic average roughness of the knives and classifying them according to the manufacturer's specification limits. Because Convolutional Neural Networks require a substantial amount of labeled data for training, this technique demands significant time and skilled personnel to annotate the instances. To address this issue, the method of Semi-Supervised Learning is applied. This approach serves as a compromise between the established machine learning methods of supervised and unsupervised learning by utilizing a small labeled dataset alongside a larger unlabeled dataset. The methodology aims to learn from unlabeled data while keeping the labeling costs as low as possible. In this application, we employ the Semi-Supervised Learning method called top-k pseudo-labeling, where the k most confident predictions of the model are assigned the model's predicted labels. Furthermore, we determine the optimal hyperparameters for the proposed algorithm and evaluate its performance from both an inductive and transductive perspective. Additionally, the limitations of the proposed algorithm are discussed, along with explanations for these constraints. Lastly, possible improvements and avenues for further investigation are presented.

Keywords: Semi-Supervised Learning, Convolutional Neural Network, self-training, pseudo labeling, production, Deep Learning.

1. Introduction

Ensuring the quality of manufactured components is of paramount importance in manufacturing processes. Since many components cannot be inspected non-destructively using conventional methods, the demand for alternative testing techniques is continuously increasing. Deep learning approaches provide a viable solution for non-destructive quality evaluation. These methodologies have demonstrated strong performance in recent years and are subject to continuous development. One prominent approach involves the application of Convolutional Neural Networks (CNN). This specific network architecture eliminates the need for manual feature extraction as a preprocessing step. Instead, these networks accept raw images as input. Through the utilization of convolutional layers, the feature extraction is performed autonomously by the network. This approach not only simplifies the workflow but also provides the neural network with complete spatial information, rendering the feature extraction process inherently trainable.

Although CNNs achieve excellent results in image classification tasks, they require substantial amounts of data for the training process. This data dependency represents a limiting factor of the technology, as every data point must be manually labeled by skilled personnel, a process that is both time-consuming and cost-intensive. To mitigate the reliance on manual labeling, models

can be trained utilizing the methodology of Semi-Supervised Learning (SSL). In this work, we evaluate the SSL method known as top-k pseudo-labeling. This methodology initially employs a supervised learning approach to train a base model. The base model is first trained on a labeled dataset and subsequently generates predictions for the unlabeled dataset. Following this, the instances for which the model exhibits the highest prediction confidence are treated as labeled data and incorporated into the subsequent training iteration.
In this project, we use an image dataset comprising 49500 images of fine-ground surfaces, which are classified according to their arithmetic average roughness value $R_a[\mu m]$. The dataset is stratified across three distinct classes. The manufacturer's specifications serve as the established classification boundaries.

2. Background and Related Work

One of the most influential studies in the field of SSL is the work presented by Yarowsky (1995). In this research, Yarowsky proposes a novel SSL algorithm for the sense disambiguation of words in English texts. While the performance of the proposed algorithm was comparable to supervised learning algorithms, the requirement for labeled data was significantly reduced. This was achieved by manually labeling only a small subset of the data and utilizing the remainder as an unlabeled dataset. Subsequently, a supervised learning algorithm is trained on the labeled dataset to predict the labels for the unlabeled data. The instances for which the model exhibits the highest confidence are then incorporated into the labeled dataset, using the model's predictions as their labels. This methodology was designated the Yarowsky Algorithm by Abney (2004), who concurrently modified the algorithm into a mathematically rigorous format. These modifications constitute the generally accepted scientific definition of the Yarowsky Algorithm.

2.1. *Taxonomy of SSL*

van Engelen and Hoos (2020) provide a detailed typology that categorizes SSL algorithms. Within the context of SSL, it is essential to distinguish between inductive and transductive algorithms. Whereas inductive methods are designed to develop a generalized model capable of predicting new data points, transductive methods do not aim for generalization. Instead, transductive methods focus exclusively on predicting the labels of the specific unlabeled data within the given dataset. In general terms, inductive methods share characteristics with supervised learning, while transductive methods align more closely with unsupervised learning. Because we evaluate a specific application of SSL, the algorithm discussed in this publication can be analyzed from both an inductive and a transductive perspective. Since the current algorithm utilizes a supervised base learner within a recursive loop, it can be classified as a wrapper method, as defined by Culp and Michailidis (2008). Furthermore, the algorithm falls under the category of self-training methods. Additionally, the presented approach employs the technique of pseudo-labeling, which treats the model's own predictions as definitive labels, in accordance with Triguero et al. (2015).

2.2. *CNNs*

CNNs are widely utilized in image, video, and signal recognition, as well as in recommender systems and natural language processing Radiuk (2017). These networks exhibit excellent performance in image classification tasks, primarily because their architecture is biologically inspired by the visual processing mechanisms of the brain Frochte (2018). A neural network is classified as convolutional if its architecture includes at least one convolutional layer Aggarwal (2018). These convolutional layers perform feature extraction, which is traditionally executed as a separate preprocessing step. In conventional feature extraction, a filter is applied to an image to manipulate its frequency spectrum. This traditional approach is inherently limited to the specific features manually selected by the user, meaning the overall performance of the model heavily depends on the chosen feature extractor. In contrast, within convolutional layers, the images are convolved with a predefined number of filters. Because the weights of these filters are trainable parameters, convolu-

tional layers automate the feature extraction process and independently identify the most relevant features for the specific application. Alongside convolutional layers, the architecture typically incorporates MaxPooling and dropout layers. For a comprehensive description of all layer types, we refer to Aghdam (2017) and Aggarwal (2018).

2.3. *Related Work*

Because SSL addresses the challenge of acquiring labeled data, it is predominantly employed in image classification tasks. A highly relevant study to this publication was conducted by Brüggemann et al. (2021), where SSL is applied to Feed Forward Neural Networks to predict the surface roughness of the identical type of chef knives analyzed in our research. Their findings indicate that SSL exhibits weaker performance when the training process includes batches with a low number of instances. Another related investigation is the research by Guenther et al. (2021). They conducted a parameter study focusing on the classification of the exact same image dataset utilized in the present publication. Their work serves as foundational research for this project, as it provides the established network architecture for the classification task and offers a baseline metric comparison for supervised learning approaches. Additionally, the classification of manufacturing defects on steel surfaces utilizing a CNN based on SSL by Gao et al. (2020) represents further relevant prior research.

3. Methodology

This section provides an explanation of the data used for the classification task, as well as the model architecture employed for the experimental setup. Furthermore, the proposed SSL algorithm is detailed.

3.1. *Dataset*

In this publication, we aim to predict the arithmetic average surface roughness (R_a value) of 851 fine-ground chef knives. To achieve this, images of the surfaces were captured under controlled conditions, and the R_a value of each surface was measured tactilely using a profilometer, as described by Hinz et al. (2019). Because 851 samples are insufficient to adequately train a CNN, each original picture ($1250px \times 550px$) is sliced into 275 smaller images ($50px \times 50px$) following the procedure established by Guenther et al. (2021). This process is visualized in figure 1.

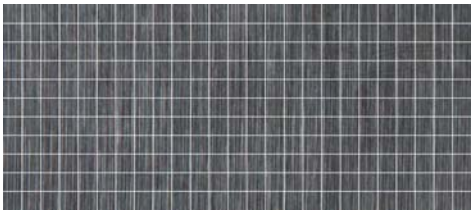

Fig. 1. Enlarging the dataset by slicing a picture in 275 images

This approach expands the total dataset to 234025 images. Because the dataset exhibits class imbalance and the subsequent training process requires substantial computational resources, we restrict the analyzed dataset to a subset of 49500 instances. These images are classified based on their surface roughness, with the R_a value serving as the target variable. The classification is executed according to the manufacturer's specification limits. These limits, along with a comprehensive summary of the utilized dataset, are presented in table 1.

Table 1. Overview of the dataset's specifications

Class	R_a specification limits	Amount of knives	Amount of cropped images	Analyzed subset
0	$< 0.13\,\mu$m	281	77275	16399
1	$0.13 - 0.21\,\mu$m	448	123200	16486
2	$> 0.21\,\mu$m	122	33550	16615

3.2. *Model Architecture*

In this study, we perform a classification task utilizing a CNN. As mentioned previously, the ar-

chitecture of the model was established by Guenther et al. (2021). Consequently, we employ a CNN comprising three convolutional layers and no dense layers. Following each convolutional layer, 2×2 MaxPooling and a dropout rate of 50% are applied. Each convolutional layer contains 128 kernels with a size of 3×3. The network is trained using a batch size of 1024 and the Adam optimizer. Since a minimum of 400 epochs is recommended, the training process is extended to 1000 epochs to ensure full convergence of the model. A visualization of the architecture can be seen in figure 2. Because the labeled dataset in this approach is significantly smaller compared to traditional supervised learning, an early stopping mechanism is implemented to prevent overfitting. The model is optimized using the categorical cross-entropy loss, defined by Goodfellow et al. (2016) as:

$$L = -\sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i) \tag{1}$$

Within the convolutional layers, the ReLU activation function is applied, which sets negative values to zero. In the fully connected layer, a softmax activation function is employed to convert logits into probabilities. The output of the softmax function subsequently serves as the model's confidence score for each class. According to Frochte (2018), the softmax function is defined as:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2}$$

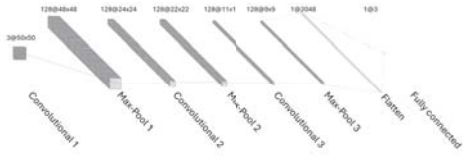

Fig. 2. Visualization of the model's architecture with all used layers except dropout between pooling- and convolutional layers. Input dimensions are listed above each layer.

3.3. *Idea and Algorithm*

The proposed algorithm is categorized as a wrapper method, wherein a recursive loop encompasses a supervised base learner. As previously established, this approach constitutes a pseudo-labeling method. Such techniques utilize the model's predictions and propagate them as the true labels for the respective data instances. Pseudo-labeling methods can generally be divided into confidence threshold methods and instance threshold methods. In this study, we employ an approach where a fixed number of data points is labeled during each iteration of the SSL algorithm. This specific methodology is defined as top-k pseudo-labeling by Jiang and Sun (2022). The proposed algorithm utilizes a small labeled dataset alongside a significantly larger unlabeled dataset. Initially, the supervised base learner is trained on the labeled data. Subsequently, the model predicts the labels for the entire unlabeled dataset. The k instances for which the model exhibits the highest confidence are then integrated into the labeled dataset, utilizing the respective predictions as their labels. To maintain stratification across the three classes, the dataset is uniformly expanded by $k/3$ instances per class. Finally, a new model is trained on this augmented dataset, and the entire procedure is repeated. Through this method, the labeled dataset is iteratively expanded by k instances per cycle.

4. Experimental Results

The algorithm utilized in this study features two primary hyperparameters: the size of the initial labeled dataset and the number of instances added to the labeled dataset per iteration, denoted as k. To analyze their impact, both hyperparameters are evaluated individually. First, to assess the dimension of k, we train the supervised base learner solely on the initial labeled dataset and predict the unlabeled data. Subsequently, these predictions are separated by class and sorted in descending order of confidence. The indices of misclassified predictions are then identified for each class. A new list is generated from these indices, containing the cumulative sum of misclassifications across all three classes for all possible indices.

Algorithm 1 Pseudo Code for Top-k Pseudo Labeling

Require: unlabeled dataset U,
initial labeled dataset L

while $U \neq \emptyset$ **do**
  $new\_model.compile()$
  $model.train(L)$
  $predictions = model.predict(U)$
  $predictions.sort(reverse = True)$

  $Pseudo\_Labels = argmax(predictions)$
  for i in range(k) **do**
    $L.append(U[i], stratified = True)$
  end for
  for i in range(k) **do**
    $U.remove(U[i])$
  end for
end while

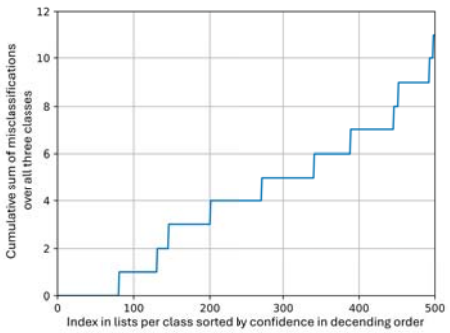

Fig. 3. cumulative sum of misclassifications across all three classes as a function of the list index

This provides an overview of the number of data points receiving an incorrect pseudo-label for each value of $k/3$. A visualization of this misclassification list near the origin is shown in figure 3. The graph is cropped, as only the range without misclassifications is relevant for this research.
Figure 3 indicates that the first misclassification occurs at index 81 in one of the three classes. Consequently, a misclassification in the initial SSL iteration is expected for stratified top-k pseudo-labeling when $k \geq 243$. Notably, only the first pseudo-labeling iteration has been investigated thus far, and the misclassification indices change with each model training due to random bias. Furthermore, the index of the first misclassification is estimated to decrease as the number of SSL iterations increases. Therefore, the subsequent analysis of the initial labeled dataset size is conducted using an estimated value of $k = 60$, adding 20 instances per class in each iteration. To determine the optimal initial training dataset size, the number of successful SSL iterations is evaluated while varying this size. This parameter is varied from 10% to 80% in increments of 10%. To provide a statistical overview of process stability, each configuration is executed three times. The results of this experiment are depicted in figure 4.

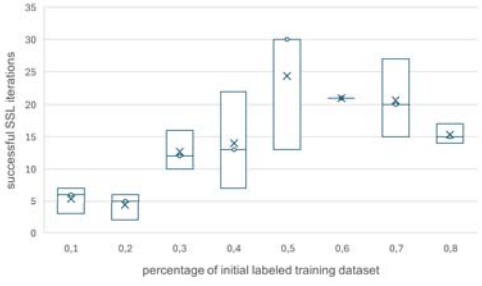

Fig. 4. Evaluation of the initial dataset size. Number of successful SSL iterations without misclassification over the percentage size of the initial training dataset in relation of the total dataset.

As illustrated in figure 4, the achieved number of SSL iterations varies even within identical configurations. This indicates that the SSL process on this dataset lacks stability, an assumption that will be discussed further. Despite this unpredictability, figure 4 reveals an increasing trend in successful SSL iterations for initial labeled dataset sizes between 10% and 50%, followed by a decreasing trend between 50% and 80%. Because an initial dataset size of 50% yielded the highest number of SSL iterations, this proportion is selected to further evaluate the value of k. The evaluation of k is conducted analogously to the previous assessment, with the initial training dataset size

fixed at 50%. The performance metric remains the number of successfully executed SSL iterations. The tested values for k are 30, 60, 90, 120, 150, and 180, corresponding to 10, 20, 30, 40, 50, and 60 instances added per class per iteration, respectively. To ensure stability evaluation, three CNNs are trained per configuration. The results are presented in figure 5.

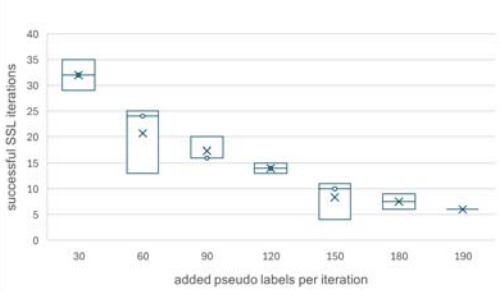

Fig. 5. Evaluation of different values for k. Number of successful SSL iterations without misclassification over the number of instances added to training data per iteration

This experiment also demonstrates variance in the outcomes; however, the underlying trend remains interpretable. As anticipated, the number of successful SSL iterations decreases as the value of k increases. This behavior is attributable to the pseudo-labeling of less confident predictions at higher values of k. Because SSL primarily aims to reduce the dependency on manually labeled data, evaluating the total number of correct pseudo-labels is essential. Consequently, the average number of successful iterations from figure 5 is used to calculate the mean number of correct pseudo-labels per configuration. The total average of correct pseudo-labels across the defined parameter space for k is illustrated in figure 6.
Because $k = 120$ yields the highest number of correct pseudo-labels without any misclassifications, it is identified as the optimal parameter value for k.

5. Evaluation

In this chapter we evaluate the model's performance in an inductive and transductive way. For

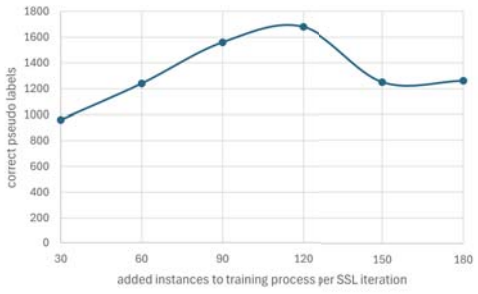

Fig. 6. Amount of correct pseudo labels depending on the number of instances added to training dataset each SSL iteration

the inductive analysis of the model's performance, the two metrics of accuracy and loss are evaluated during the SSL process. However, the accuracy and the loss of the models do not change significantly. This also happens after the occurrence of the first misclassification which leads to further misclassifications in later SSL iterations. The curve of the accuracy does not change very much before and after the first misclassification which means that misclassified data by the model in the training dataset does not affect the accuracy. This aspect will be discussed in the next chapter. However the proposed SSL settings lead to an accuracy of 83.7% which is only 2.3% less than the supervised learning counterpart while reducing the need of labeled data by 50%. In a transductive perspective, the best model was able to create 1800 correct pseudo labels. This leads to the conclusion, that a top-k pseudo labeling SSL model can assist the manual labeling process. Since a misclassified instance in the training process leads to further misclassified instances, a SSL setup can not be used to label the entire dataset.

6. Discussion of Results

A primary issue to discuss is the high variance in successful SSL iterations for identical parameter configurations. This aspect is crucial, as single parameter evaluations may yield unrealistic trends. This variance can be attributed partly to differing training behaviors caused by the random initialization of the model's weights and biases. Defin-

ing a fixed seed for initialization could mitigate this instability. Another contributing factor is the weakened performance of SSL algorithms when training batches are underfilled, as suggested by Brüggemann et al. (2021). This occurs because gradients are computed as an average across the data points within each batch. An underfilled final batch results in a gradient calculated from significantly fewer instances, leading to a noisier gradient average compared to a fully populated batch of 1024 data points. This noise is particularly problematic for SSL tasks, as the underfilled batch typically dictates the final gradient calculation of an epoch. Consequently, the final weight adjustment prior to predicting the unlabeled data relies on this noisy gradient. While discarding incomplete batches easily resolves this in standard supervised learning, it is counterproductive for SSL, which fundamentally relies on iteratively maximizing the use of available labeled data.

From an inductive perspective, it is notable that the SSL approach did not improve the model's overall accuracy. This indicates a requirement for a larger volume of correctly pseudo-labeled data points to enhance performance. Furthermore, the model's accuracy remains remarkably stable even following the introduction of a misclassification. This stability might stem from the limited precision of the tactile measurement system, compounded by the assumption that the R_a value remains constant across the entire knife surface. While potentially problematic for data points near classification boundaries, this hypothesis requires further investigation. Finally, the occurrence of misclassifications during the pseudo-labeling process must be addressed. As explained by Arazo et al. (2020), this phenomenon results from confirmation bias when employing hard pseudo-labels in SSL. Confirmation bias occurs when the model reinforces mistakenly learned rules that deviate from the underlying data distribution. To mitigate this, the implementation of soft labels or mixup augmentation is proposed.

Regarding the specific application of chef knife manufacturing, the implications of this research are highly relevant. The traditional determination of surface quality through tactile profilometry represents a significant bottleneck in the production line. The successful application of the proposed SSL methodology demonstrates that optical and non-destructive evaluations can be integrated even when annotated data is highly limited. The technical consequences of the algorithm's current performance suggest a hybrid integration strategy. Instead of replacing conventional quality control entirely, the system functions optimally as an automated preliminary filter. Knives that are predicted with high confidence to be within or outside the manufacturer's specification limits can be processed immediately. Conversely, components with uncertain classifications are routed for tactile measurement. This approach significantly increases the production throughput while maintaining the stringent technical requirements of surface roughness evaluation. In conclusion, the algorithm holds significant potential as an assistive tool for skilled personnel. Because class 1, which represents knives meeting the production specifications, is comparatively challenging for the model to learn, the algorithm is expected to maintain robust performance primarily in identifying clearly compliant or non-compliant components.

7. Conclusion and Outlook

In the present paper, an SSL wrapper method utilizing a supervised CNN base learner is proposed. The algorithm is based on the principle of top-k pseudo-labeling. This configuration involves two primary hyperparameters: the size of the initial labeled dataset and the number of instances incorporated per iteration. These parameters were optimized to maximize the yield of correct pseudo-labels. Utilizing an initial labeled dataset of 50% and $k = 120$, an inductive accuracy of 83.7% is achieved. From a transductive perspective, 1800 instances are labeled correctly without any misclassifications. Furthermore, identified limitations of the proposed algorithm provide directions for future investigations. Specifically, subsequent research should analyze the impact of underfilled batches on gradient calculations and the resulting weight updates. Additionally, employing higher-precision instruments for surface roughness determination, such as confocal microscopy, could

enhance ground truth accuracy. Algorithm performance could be further improved by applying soft pseudo-labels and mixup augmentation techniques. Finally, optimizing the CNN architecture through hyperparameter tuning tailored for small datasets in a semi-supervised context, alongside exploring ensemble methods for the base learner, remain promising avenues for future research.

References

Abney, S. (2004). Understanding the yarowsky algorithm. *Computational Linguistics 30*(3), 365–395.

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Textbook*. Cham: Springer International Publishing AG.

Aghdam, H. H. (2017). *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*. Springer eBook Collection Computer Science. Cham: Springer.

Arazo, E., D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness (2020). Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, Piscataway, NJ, USA, pp. 1–8. IEEE.

Brüggemann, D., M. Hinz, and S. Bracke (2021). An application of semi-supervised to production data. In B. Castanier, M. Cepin, D. Bigaud, and C. Berenguer (Eds.), *Proceedings of the 31st European Safety and Reliability Conference (ESREL 2021)*, Singapore, pp. 1662–1669. Research Publishing Services.

Culp, M. and G. Michailidis (2008). An iterative algorithm for extending learners to a semi-supervised setting. *Journal of Computational and Graphical Statistics 17*(3), 545–571.

Frochte, J. (2018). *Maschinelles Lernen: Grundlagen und Algorithmen in Python*. München: Hanser.

Gao, Y., L. Gao, X. Li, and X. Yan (2020). A semi-supervised convolutional neural network-based method for steel surface defect recognition. *Robotics and Computer-Integrated Manufacturing 61*, 101825.

Goodfellow, I., A. Courville, and Y. Bengio (2016). *Deep learning*. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press.

Guenther, L. H., M. Hinz, and S. Bracke (2021). Cnn based analysis of grinded surfaces. In B. Castanier, M. Cepin, D. Bigaud, and C. Berenguer (Eds.), *Proceedings of the 31st European Safety and Reliability Conference (ESREL 2021)*, Singapore, pp. 1026–1033. Research Publishing Services.

Hinz, M., M. Radetzky, L. Hannah Guenther, P. Fiur, and S. Bracke (2019). Machine learning driven image analysis of fine grinded knife blade surface topographies. *Procedia Manufacturing 39*, 1817–1826.

Jiang, Y. and H. Sun (2022). Top-k pseudo labeling for semi-supervised image classification. *International Journal of Data Warehousing and Mining 19*(2), 1–18.

Radiuk, P. M. (2017). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. *Information Technology and Management Science 20*(1).

Triguero, I., S. García, and F. Herrera (2015). Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems 42*(2), 245–284.

van Engelen, J. E. and H. H. Hoos (2020). A survey on semi-supervised learning. *Machine Learning 109*(2), 373–440.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In H. Uszkoreit (Ed.), *Proceedings of the 33rd annual meeting on Association for Computational Linguistics* -, Morristown, NJ, USA, pp. 189–196. Association for Computational Linguistics.