

16th CIRP Conference on Intelligent Computation in Manufacturing Engineering, CIRP ICME '22, Italy

Introduction to deep degradation metric in smart production ecosystems

Yeremia Gunawan Adhisantoso*, Quy Le Xuan, Christoph Kellerman, Marco Munderloh, Jörn Ostermann

Institut für Informationsverarbeitung, Appelstraße 9A, Hannover 30167, Germany

* Corresponding author. Tel.: +49 511 762-19588; E-mail address: adhisant@tnt.uni-hannover.de

Abstract

With the advent of Industry 4.0, more data is exploited to improve efficiency in production and to enable a cost-effective maintenance approach called predictive maintenance. In a production ecosystem, assets are maintained based on their corresponding internal condition. Often, there is no known ground truth or label for the internal condition of the components, especially for high-dimensional data. Furthermore, the initial degradation condition of the assets differs from each other. We present a novel approach to learn a degradation metric implicitly (with minimal information). The approach can take the initial condition of the asset into consideration.

© 2022 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the scientific committee of the 16th CIRP Conference on Intelligent Computation in Manufacturing Engineering.

Keywords: computer vision; predictive maintenance; prognostic health management;

1. Introduction

In the manufacturing world, the role of maintenance becomes increasingly important. When well performed, it can reduce downtime and production costs. In addition, well maintained assets guarantee the quality of the goods manufactured. An asset can be utilized until its end-of-life (EOL) is reached. EOL describes the time index at which the asset health falls below or degradation raises above a certain threshold. The time until EOL is reached is described as remaining useful life (RUL). Two well-known maintenance methods [1] are reactive maintenance and preventive maintenance. With reactive maintenance, an asset is used until the end-of-life is reached, allowing maximum utilization of an asset. For critical applications, downtime caused by EOL is undesirable. On the other hand, an asset is maintained periodically based on preventive maintenance to prevent possible downtime or possible sudden breakdown. Assets maintained in this manner are often replaced or maintained despite their good condition. This results in under-utilization of the asset and higher cost.

A modern approach to Prognostic Health Management (PHM) is predictive maintenance. It leverages the use of big data coming from sensors to analyze the internal condition of an asset. When a certain condition is observed, the asset will be maintained. Furthermore, remaining useful life (RUL) is estimated at every time point, providing information when the asset should be ideally maintained. Essentially, predictive maintenance maximizes utilization while minimizing downtime.

In our case, we focus on predicting the RUL of assets used in glass syringe production, specifically in the print screen step where a certain layout is printed on top of syringe surface. The assets mentioned do not have any sensor attached. Therefore, direct condition assessment is not possible. We monitor the condition of the assets indirectly by assessing the inspection image of produced syringes. For high dimensional data, inferring the internal condition is a complex task. In addition, ground truths or labels describing the degradation or health index of the asset at each time point are not always available.

In this paper, we propose Deep Degradation Metric (DDM) method that implicitly learns the degradation index by

exploiting the property of the end-of-life prediction. Our contribution in this work can be summarized as follows:

- Unsupervised RUL prediction on high-dimensional data
- Estimation of a degradation index of an asset by exploiting the property in the end-of-life prediction.
- Automatic relevant determination of features that play an important role in determining degradation or health index.

Nomenclature

PHM	Prognostic Health Management
EOL	End-of-Life
RUL	Remaining Useful Life
ICM	Indirect Condition Monitoring
DI	Degradation Index
DDM	Deep Degradation Metric
k-NN	k-Nearest Neighbour
DML	Deep Metric Learning
ROI	Region of Interest
BCE	Binary Cross Entropy
MSE	Mean-squared Error
ReLU	Rectified Linear Unit

2. Related works

Many works have been done in the field of PHM. Saxena et al. [2] describe that unlike any other future behavior predictions, EOL prediction has unique properties: a prediction threshold exists and the trend is monotonic in general. Si et al. [3] described comprehensively models that can be used to predict the RUL based on the observed condition, the historical data of similar assets or degradation index.

In the field of computer vision, deep convolutional neural networks [4]–[6] have had a great success in solving task such as image classification. There are two well-known architectures: ResidualNet (ResNet) [4] and EfficientNet [5], [6] comprised of predefined blocks such as Basic Block, Bottleneck Block and Inverted Residual Bottleneck. This simplifies the scaling of the model to a deeper model, achieving better performance. Additionally, skip or residual connections have been integrated to solve the vanishing gradient problem that plagues deep neural network models. Both ResNet and EfficientNet belong to the so-called foundation models [7] that serve as the backbone of other neural networks for various tasks.

In PHM or RUL prediction, the features of an assets with comparable health index are similar, thus k-Nearest Neighbor (k-NN) [8], [9] can be employed to predict the health index based on the similarity of the features of an asset to the features in the database. For high dimensional data such as image, a deep neural network can be trained to measure the similarity between features of assets. One specific field in deep learning, called deep metric learning (DML) [10], [11], attempts to map the datapoints or features to an embedding space, where similar datapoints are located near each other and dissimilar datapoints apart from each other. Contrastive Loss [12] pulls together similar datapoints while dissimilar datapoints are pushed away from each other. On the other hand, Triplet Loss [13] tries both at the same time. Given three data, anchor, positive (data with

similar label) and negative (data with different label), it tries to minimize the distance between anchor and positive data while maximizing the distance between anchor and negative data. Unfortunately, labels are required for DML methods and such information is expensive to generate and often unavailable in manufacturing process. In our work, we extend the DML by exploiting special property of RUL prediction. It requires no ground truths or labels corresponding to internal condition.

3. Method

3.1. Problem scenario

Our goal is to predict the RUL in the absence of ground truth. Using the time index directly as the label is not suitable due to various reasons. First, the initial degradation index of an asset after maintenance or replacement are different, even with quality control. By using time index as the label, we infer that the degradation of all assets is equal at the beginning. Then there is another challenge, which is the trend of the degradation index. Time index increases linearly from 0 to n where n is the time index when the process ends (or when a certain threshold is reached). This will result in the assumption that the trend is linear while in real production the degradation trend is unknown.

3.2. Deep Degradation Metric

One property in EOL prediction is particularly interesting for us. Without maintenance, the condition of an asset degrades until it loses a degree of functional capability (including complete failure), therefore the trend in EOL prediction is monotonic as it goes from healthy condition to degraded condition [2]. Based on this property, we design our loss function and transform the time index so that we can use it as a pseudo label.

Assume that for a given asset, the corresponding true health index develops over time as shown in Fig. 1. The internal condition is unknown to us. Given two data points A and B corresponding to an asset, we can qualitatively assume that the health index at time index t_A is lower than at time index t_B . We formulate this problem by predicting the order of two images, such that given two data points which has the higher time index. Therefore, we transform our problem into a classification problem by transforming the time index:

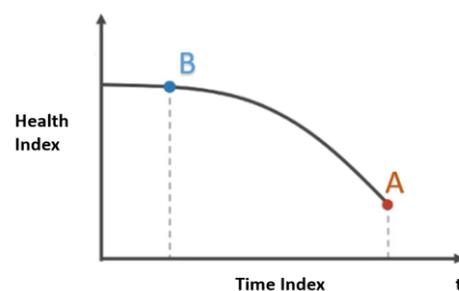


Fig. 1. Health index of an asset over time. Given two measurement points at time index t_A and t_B , we can assume the health index at t_A is lower than at time index t_B .

$$y_{A,B} := \begin{cases} 0, & \text{if } t_A < t_B \\ 1, & \text{otherwise.} \end{cases} \quad (1)$$

$y_{A,B}$ is the pseudo label for time index t_A and t_B . Changing the order of A and B will result in different value of y . The model predicts the order by predicting the degradation index m_A and m_B for data points (in this case images) A and B , respectively. The predicted degradation indices are then transformed into predicted pseudo label $\hat{y}_{A,B}$:

$$\hat{y}_{A,B} := \begin{cases} 0, & \text{if } m_A < m_B \\ 1, & \text{otherwise.} \end{cases} \quad (2)$$

Both transformations are applied only for the training and the validation phase. For the inference, we predict the degradation index for each image, removing the need of second image. If the mean-squared error is used as the loss function, the value -1 can be used instead of 0 in both equation (1) and (2). It is called pseudo label because it captures the interaction between two datapoints qualitatively in the absence of true label or ground truth.

We do not call it true label because it has the probability of being correct. For each pseudo label $y_{A,B}$, there exists a corresponding true label $\tilde{y}_{A,B}$ that is unknown to us. We measure the internal condition indirectly and the quality of the produced syringe is influenced by other external factors. The probability that the pseudo label is equal to the true label given time given the time indices of the corresponding data points, t_A and t_B ,

$$p_{y=\tilde{y}} := p(y_{A,B} = \tilde{y}_{A,B} | t_A, t_B) \quad (3)$$

becomes more random as the time index difference between t_A and t_B becomes smaller. Thus, we can describe the probability of the pseudo label being correct is proportional to the function of the time index difference between data A and data B:

$$(p_{y=\tilde{y}}) \log(p_{y=\tilde{y}}) \sim f(|t_A - t_B|) \quad (4)$$

We do not set a minimum distance between t_A and t_B as it is a hyperparameter that is directly optimized. Through the experimental results (see Fig. 7), we show that DDM can learn the noise in the production automatically. **In addition, pairs with lower distance in time index will serve as good training**

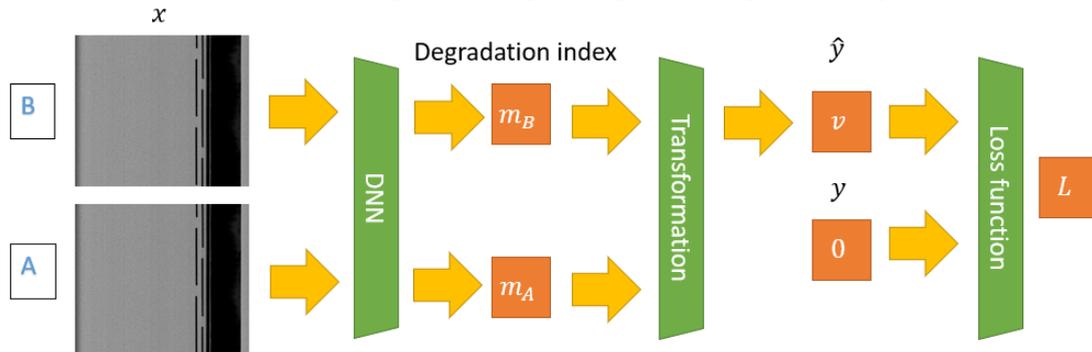


Fig. 2. Deep Degradation Metric framework. The core idea of our DDM framework is to predict which image was taken later in the production. The input is a pair of images of the same sequence. For each image, the degradation index m_A and m_B are predicted. The predicted degradation indices are then transformed into the predicted pseudo label \hat{y} . The time indices of the images are transformed into pseudo label y . The loss function is either binary cross-entropy or mean-squared error between \hat{y} and y .

samples. When both datapoints are located near each other in terms of time index, the difference in degradation between data becomes less apparent. It enables the neural network to better learn subtle changes in the data and how it changes over time. Since our labels have the probability of being correct (Eq. 4), the goal of deep degradation metric is not to achieve hundred percent accuracy in classification problem but to minimize the overfit between training and validation and to learn the degradation index implicitly.

Lastly, pseudo label transformation is only allowed when both datapoints belong to the same sequence. This is because the monotonicity property holds true as long as both data points belong to the same sequence. We cannot compare the health index of two different assets at the same time index nor make any assumption.

The proposed framework of Deep Degradation Metric can be seen in Fig. 2. During training, pairs of data x_A and x_B , the neural network will output degradation indices m_A and m_B . Similar to the time index, the degradation indices are transformed into predicted pseudo label \hat{y} . At the end of the pipeline, the loss function such as binary cross entropy L_{BCE} or mean-squared error L_{MSE} is computed:

$$L_{BCE} = -\frac{1}{N} \sum y_{A,B} \cdot \log \hat{y}_{A,B} + (1 - y_{A,B}) \cdot \log(1 - \hat{y}_{A,B}) \quad (5)$$

$$L_{MSE} = \frac{1}{N} \sum (y_{A,B} - \hat{y}_{A,B})^2 \quad (6)$$

where N is the number of datapoint pairs in one mini-batch. The loss is computed based on pseudo label y and predicted pseudo label \hat{y} . The degradation index is learned implicitly with by utilizing pseudo label transformation. One input is sufficient at inference time to predict the degradation index.

4. Experiment

4.1. Dataset

We performed our experiments on image data, comprised of high-resolution inspection image of produced syringes (see Fig. 3). All image data is monochrome. Using the screen-printing method, a layout is printed on top of the surface of a syringe. Our dataset consists of 4 sequences of grayscale inspection images. In total, our dataset is comprised of 46,389 inspection images. One sequence ends when the maintenance

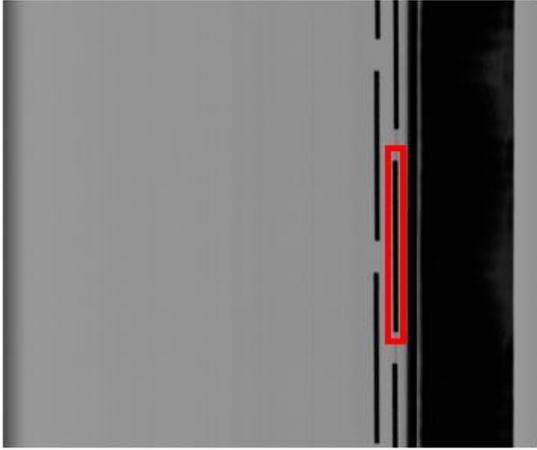


Fig. 3. The inspection image in the syringe production. The layout or ROI occupies a small portion of the image and thus can be extracted from the image using a template matching algorithm.

is applied (in this case when the machine is opened). The first sequence contains defect and therefore used in the test phase. By using a sequence that contain defects for the test, we can analyze the behavior of the model in the case of anomaly or defect. Sequence 2 to 4 are used for both training and validation phase.

4.2. Preprocessing

To minimize the influence of external factor, the region of interest (ROI) is extracted (see Fig. 4) using a template matching method. Then, the extracted ROI becomes the input of the neural network x . Furthermore, by extracting ROI from the original image we greatly reduce the memory requirement for the neural network and reduce the computation cost. Thus, we decrease training time as the batch size can be increased.

Because the glass syringe is transparent, the printed image on the bottom side of the syringe is captured in the inspection image. To mitigate this problem, the camera is angled so that the layout on the bottom side and the top side do not overlap with each other. In the ROI extraction, only the layout corresponding to the top side and focused by the camera is extracted and used as the input as seen in Fig. 5.

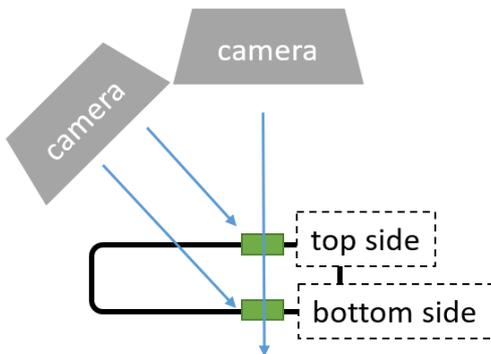


Fig. 4. Inspection system in the syringe production. Because the glass is transparent, the printed image on the bottom side of the syringe is captured by the camera. Thus, the printed images on the top and bottom side overlap with each other. The camera is tilted at a certain angle to separate the printed images from each other in the inspection image.

4.3. Training

For each mini-batch during training phase, pairs of image data are randomly selected. Both images of a pairs must belong to the same sequence. As we discussed earlier, we do not set a minimum distance between two randomly selected images.

4.4. Architecture

The size of each ROI image is 720-by-30 pixels, which is unusual for a standard convolutional neural network architecture. Furthermore, the input is a grayscale image that consists of only one channel. Thus, we use the ResNet architecture with modified hyperparameters that is suitable for our use case. Our architecture uses slightly modified basic block of the ResNet as the building block. Rectified linear activation unit (ReLU) in the basic block is replaced with Leaky ReLU [14], [15] activation unit to allow better backpropagation in a deeper neural network architecture. Similar to the original basic block, each block reduces the spatial resolution by half. We vary the number of blocks between 6 to 9.

All of the hyperparameters except the learning rate and the learning rate policy are optimized during the training using a tool called Optuna [16]. For the learning rate and learning rate policy we follow the best practice guide from Smith *et al.* [17], [18]. Learning rate test is utilized to find the best range of learning rates. This is based on the idea that using a large learning rate will accelerate the convergence rate. However, if the learning rate is too large, the training will overfit and might cause the training to diverge. The learning rate test is done by training the neural network for one epoch. During the training, the learning rate is varied exponentially from the lowest learning rate allowed to the highest learning rate allowed. The range of learning rates is selected where the slope of the loss value is at the highest before the loss value reaches its minimum. We varied the learning rate during the training following 1cycle learning rate policy [18].

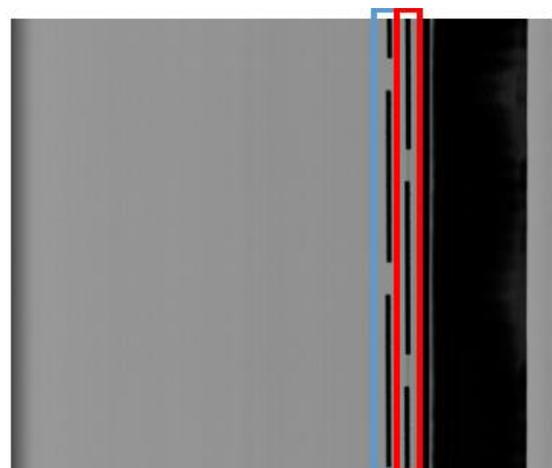


Fig. 5. With tilted camera, the bottom (blue) and the top (red) side of the printed image are separated. Only the top side part is used as the input of the neural network.

4.5. Anomaly in the dataset

The monotonic trend in the RUL prediction holds true in most of the cases. However, problem arise when the dataset contains anomalies or defects. Anomalous data has a very different degradation index in comparison to the neighboring data points as can be seen in Fig. 6. If the information about anomalous data exists, we can exclude the anomalous data or the sequence from the training as we did with sequence 1. Without label, exclusion of anomalous data is not possible. Yet, without removing the possible anomalous data, it will degrade the performance of the model. This happens because the time index does not correlate with the degradation index anymore. To remove possible anomalous data, we did multi-pass training. The idea is that our model can detect anomalous data but not perfect (see Fig. 7). After the first training, data points are classified based on the distance between itself and the actual trend. The threshold can be computed based on standard deviation or quantile. In our experiment, a quantile-based threshold performs better as the quantile-based approach is more robust against anomalies and outliers. The data classified as outlier or anomaly are then removed from the next training. We repeat this process three times or until the number of possible outliers is negligible.

5. Result

The predicted degradation index can be seen in Fig. 7 with pseudo label prediction accuracy reached at 84%. The blue line depicts the predicted degradation index at each time index. From the predicted degradation index, we can see that the production process is noisy due to various reasons. The orange line is the trend of the degradation index computed based on moving average with window size of 35. Each sequence starts from **different initial degradation index**. This reflects the

difference in the start condition of the asset at the beginning of the process. Furthermore, the developments of the degradation index are also different from one sequence to another. It can be seen by looking at the different slope value of each trend. The red line is the threshold at which the asset should be maintained. **The threshold value can be determined based on the degradation index at the end of the sequences used for training.** At the end of sequence 4, the degradation index is slightly lower compared to other sequences, signaling for a possible under-utilization of an asset. For the sequence 1, the neural network behaves as we expect given defective printed image by predicting much higher degradation index compared to the degradation index of nominal data.

6. Conclusion

We present a novel approach called Deep Degradation Metric for predicting the degradation index implicitly without

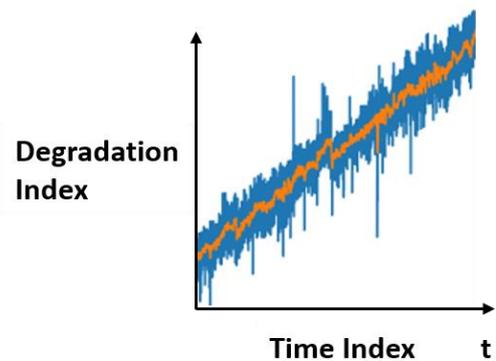


Fig. 6. The trend (orange) of the degradation index in RUL prediction is monotonic. However, the sequence might contain anomalous data, depicted as datapoints with great difference compared to the trend. Anomalous data are removed using multi-pass training to improve performance.

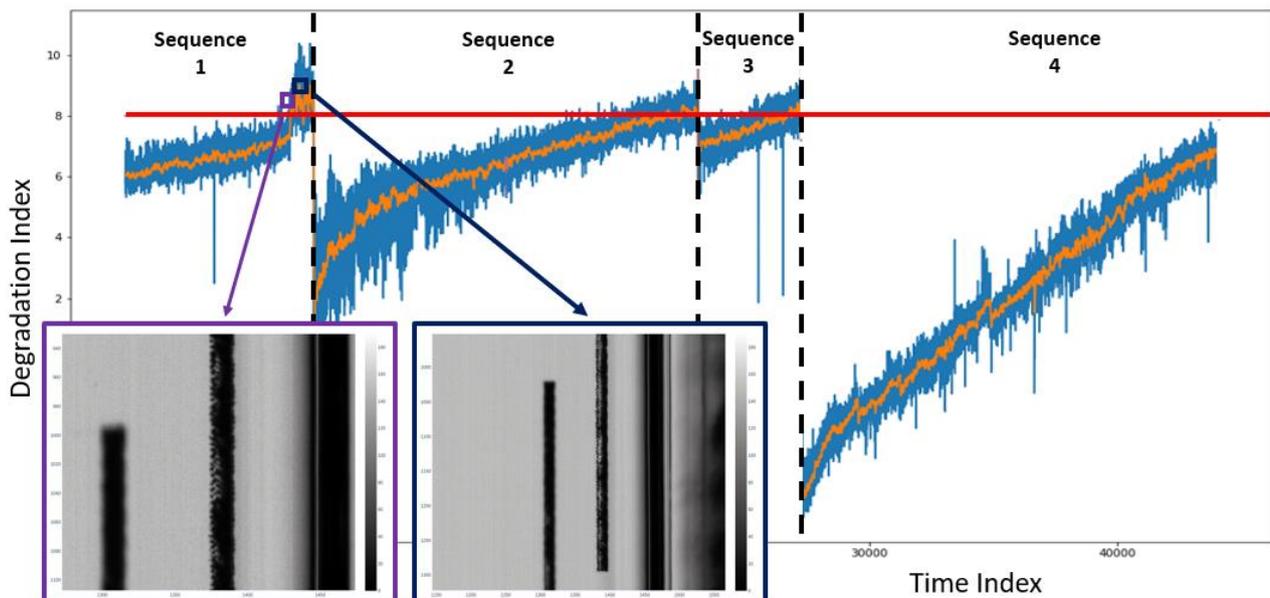


Fig. 7. The predicted degradation index of all data, including anomalous data. Sequence 2 to 4 are used for both training and validation. The neural network predicts the degradation index at each time point (blue). The trend (orange) is computed based on moving average. The threshold (red) is determined based on the degradation index at the end of the sequences used for both training and validation. The neural network behaves as expected for defective inspection image by predicting much higher degradation index compared to nominal ones, as seen in sequence 1.

ground truth and the prediction can be done at each time index. This is done by exploiting special property in the remaining useful lifetime prediction. We formulate this problem by predicting the order of the two data corresponding to the same sequence using into pseudo label. The transformation is applied to the time index and predicted degradation index. The prediction of pseudo labels reached 84% during the experiment. During the inference time, one data is sufficient to predict the corresponding degradation index. By predicting the health or degradation index of an asset at each time point, we can have a better insight of how the degradation develop over time and avoid possible under-utilization. Information of the development of the degradation can be utilized to fine-tune the production process in such a way that the degradation develops slower over time.

Acknowledgements

The authors acknowledge the financial support by the Federal Ministry for Economic Affairs and Climate Action of Germany (BWMK) in the framework of IIP-Ecosphere project (project number 01MK20006A). We would also like to thank Gerresheimer Bünde GmbH for providing the materials in this work.

References

- [1] G. Niu, *Data-Driven Technology for Engineering Systems Health Management*. Singapore: Springer Singapore, 2017.
- [2] A. Saxena et al., "Metrics for evaluating performance of prognostic techniques," in *2008 International Conference on Prognostics and Health Management*, Denver, CO, USA, Oct. 2008, pp. 1–17.
- [3] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation – A review on the statistical data driven approaches," *European Journal of Operational Research*, vol. 213, no. 1, pp. 1–14, Aug. 2011.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
- [5] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *arXiv:1905.11946 [cs, stat]*, Sep. 2020.
- [6] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," *arXiv:2104.00298 [cs]*, Jun. 2021.
- [7] R. Bommasani et al., "On the Opportunities and Risks of Foundation Models," *arXiv:2108.07258 [cs]*, Aug. 2021.
- [8] E. Fixt and J. L. Hodges, "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties," p. 11, 2022.
- [9] "An Introduction to Kernel and Nearest-Neighbour Nonparametric Regression," p. 12, 2022.
- [10] K. Musgrave, S. Belongie, and S.-N. Lim, "A Metric Learning Reality Check," in *Computer Vision – ECCV 2020*, vol. 12370, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 681–699.
- [11] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen, "Revisiting Training Strategies and Generalization Performance in Deep Metric Learning," p. 11.
- [12] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, San Diego, CA, USA, 2005, vol. 1, pp. 539–546.
- [13] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 815–823.
- [14] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv:1511.06434 [cs]*, Jan. 2016.
- [15] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," p. 6.
- [16] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *arXiv:1907.10902 [cs, stat]*, Jul. 2019.
- [17] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, USA, Mar. 2017, pp. 464–472.
- [18] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay," *arXiv:1803.09820 [cs, stat]*, Apr. 2018.