

Corrosion Image Data Set for Automating Scientific Assessment of Materials

Biao Yin¹

byin@wpi.edu

Nicholas Josselyn¹

njjosselyn@wpi.edu

Thomas Considine²

thomas.a.considine.civ@army.mil

John Kelley²

john.v.kelley8.civ@army.mil

Berend Rinderspacher²

berend.c.rinderspacher.civ@army.mil

Robert Jensen³

robert.e.jensen.civ@army.mil

James Snyder⁴

james.f.snyder.civ@army.mil

Ziming Zhang¹

zzhang15@wpi.edu

Elke Rundensteiner¹

rundenst@wpi.edu

¹ Data Science Program

Worcester Polytechnic Institute

Worcester, MA

² Weapons and Materials Research

Directorate

US Army Research Laboratory

Aberdeen Proving Ground, MD, USA

³ Weapons and Materials Research

Directorate

ARL Northeast Regional Extended Site

Burlington, MA, USA

⁴ Weapons and Materials Research

Directorate

DEVCOM Army Research Laboratory

Adelphi, MD, USA

1 Introduction

In this supplementary material we provide:

- Additional corrosion image examples
- Corrosion experimental coating stack-up information and image naming schemes
- Examples of non-expert measurements
- Model architecture pipeline diagrams and hyper-parameters
- Augmentation method and parameter details
- Additional results and conclusions of pretrained models
- Additional Grad-CAM visualization results

2 Data Set

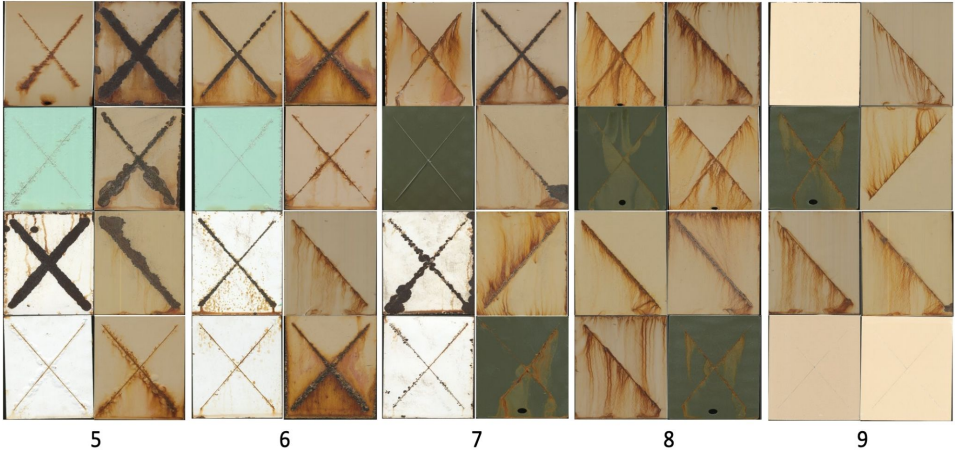


Figure 1: More corrosion image examples.

Shown here in Supplementary Figure 1 are more sample images of scribe corrosion ratings 5-9; 8 examples of each rating class are shown. Again, we highlight the variety in appearance of panels for a single rating class; images have single or double scribes, various panel background colors, different colors of corrosion, and corrosion that has not perforated the topcoat. Furthermore, we re-emphasize the small differences apparent between categories upon simple visual inspection and highlight the necessity for precise measurements of corrosion width to determine corrosion ratings. Finally, we see that in higher ratings, such as 8 and 9, corrosion can be so thin that it is hard to visually see and requires magnifying glasses to make precise measurements.

The data released in this work comes from standardized material science experiments for the purpose of materials research. With that, there exists unique stack-up configurations to each panel that we present. Present in the released data will include image names for each image in the structure of: I#_Substrate_Profile_Pretreatment_Primer_Topcoat. Each of these pieces of the stack-up are shown in Supplementary Figure 3. All proprietary product information has been removed from these image names and replaced with generic terminology provided by domain experts. We will release all 600 images upon acceptance following this naming scheme for all our images. Included in the release will be a folder of all 600 images, the data split into our 10 cross-validation folds and held-out test set for reproducibility of results, and an excel sheet of all 600 image names with their corresponding scribe corrosion rating.

In Supplementary Figure 3 we see an example of what a coating stack-up is for an image panel. It contains substrate, profile, pretreatment, primer, and topcoat layers, applied in that order. The substrate layer is the material being coated, the profile layer is the shaping, smoothing and cleaning of the material, the pretreatment layer is applied to assist in primer adhesion, the primer layer provides corrosion resistance, and the topcoat is the primary layer at risk of contamination and is of the most focus for formulation efforts to enhance agent resistance.

In Supplementary Figure 4 we see a full scribe rating table from the ASTM standard [5]. Millimeter and inch measurement ranges are shown for each of the 11 (0-10) scribe rating

categories.

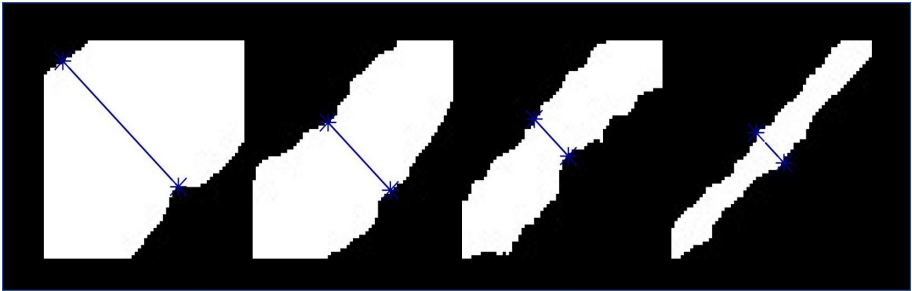


Figure 2: Sample images of non-expert measurements.

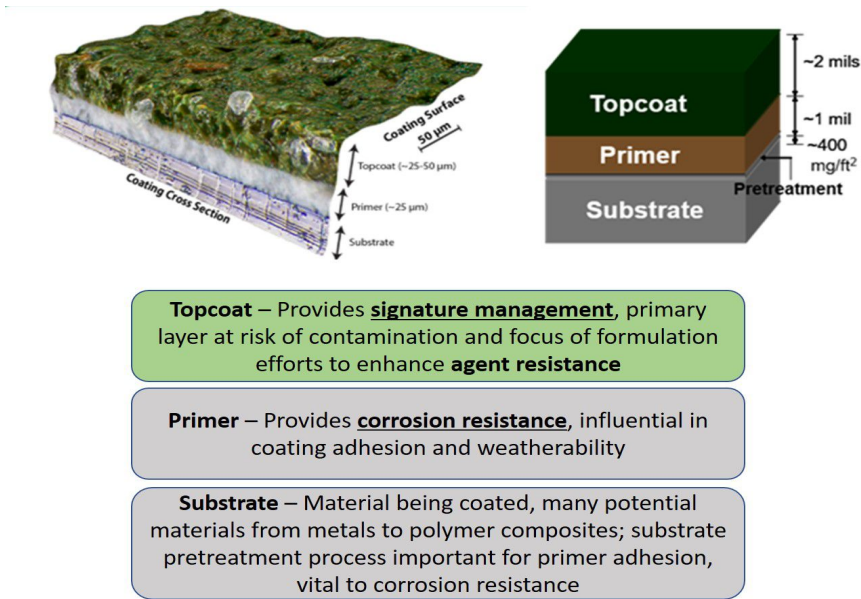


Figure 3: Experimental stack-up for corrosion testing.

3 Non-expert Study

In Supplementary Figure 2, we show examples of how corrosion is measured in the non-expert study. Non-experts first segment the areas of corrosion and then 12 equally distributed points are measured across the scribe and the conversion to mm length, and subsequently a corrosion rating, is found using the pipeline and equations in the main paper.

In this non-expert study, we crop 12 or 6 square boxes along the scribe on each image. In each box, we use computational tools to define its corrosion segmentation, shown as white areas, and draw the intersection of the box diagonal line and the white area as the corrosion pixel width of the box. We then average the width among all the boxes, convert to mm, divide by 2, and assign a corrosion rating.

| Representative Mean Creepage From Scribe | | |
|--|-------------------------|------------------|
| Millimetres | Inches (Approximate) | Rating Number |
| Zero | 0 | 10 |
| Over 0 to 0.5 | 0 to 1/64 | 9 |
| Over 0.5 to 1.0 | 1/64 to 1/32 | 8 |
| Over 1.0 to 2.0 | 1/32 to 1/16 | 7 |
| Over 2.0 to 3.0 | 1/16 to 1/8 | 6 |
| Over 3.0 to 5.0 | 1/8 to 3/16 | 5 |
| Over 5.0 to 7.0 | 3/16 to 1/4 | 4 |
| Over 7.0 to 10.0 | 1/4 to 3/8 | 3 |
| Over 10.0 to 13.0 | 3/8 to 1/2 | 2 |
| Over 13.0 to 16.0 | 1/2 to 5/8 | 1 |
| Over 16.0 to more | 5/8 to more | 0 |

Figure 4: ASTM scribe corrosion rating table.

4 Model Architectures

In this work we use five primary model architectures: ResNet-18, ResNet-50 [10], DenseNet [11], HRNet [12], and Pretext-Invariant Representation Learning (PIRL) [13]. Our experiment pipelines for all experiments shown in Table 2 are demonstrated in Figures 5 and 6. In this section we present our experiment pipelines, outline key model hyper-parameters, augmentation methods used, and augmentation parameters searched and/or found. In Table 1 we show all 9 augmentation methods used in our work, a description of what each method does to an image, and the parameters and values for each parameter we search over; the bolded values are the determined tuned values presented in the main paper.

As shown in Supplementary Figure 5, we train ResNet-18, ResNet-50, DenseNet, and HRNet models from scratch on our corrosion data with (red) and without (blue) data augmentation as well as fine-tuning pretrained ResNet-18 and ResNet-50 models with data augmentation (green). In these experiments, for all models we use: momentum 0.9, cosine learning rate scheduler with exponential warmup, SGD optimizer, and train for 2000 epochs. For ResNet-18 and ResNet-50, we use a learning rate of 0.003 and weight decay of 0.05. For DenseNet we use a learning rate of 0.0004 and weight decay of 0.05. For HRNet we use a learning rate of 0.003 and weight decay of 0.0004. For ResNet-18 we use a batch size of 64. For ResNet-50, DenseNet, and HRNet we use a batch size of 32.

As shown in Supplementary Figure 6, we not only trained ResNet with the recent self-supervised learning approach, Pretext Invariant Representation Learning (PIRL), from scratch but also applied a pretrained ResNet with the approach using ImageNet. The pretrained learned representation was transferred to our downstream corrosion assessment task for predicting expert corrosion ratings.

In part (a) of Supplementary Figure 6, PIRL trained from scratch, our final results come from the following tuned hyper-parameters: no data augmentation, learning rate 0.03, 2000 epochs, batch size 64. For its corresponding downstream task, we tuned the linear or mlp layer (added a relu activation layer) using the following tuned hyper-parameters: data augmentation, learning rate 0.5, 2000 epochs, batch size 64, cosine learning rate decay. In the

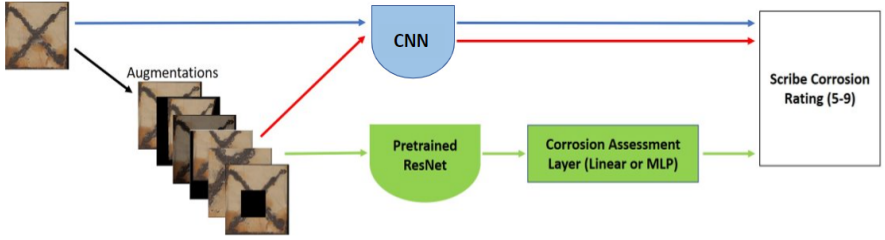


Figure 5: Experiment pipelines for ResNet-18 and ResNet-50 PyTorch-pretrained (ImageNet) and ResNet-18, ResNet-50, DenseNet, and HRNet (denoted together as "CNN") trained from scratch results. Blue: no augmentation, trained from scratch pipeline. Red: with augmentations, trained from scratch pipeline. Green: pretrained with augmentations pipeline.

pretrained PIRL, we fine-tuned the model and the downstream linear or mlp layer (add a relu activation layer) using the following tuned hyper-parameters: data augmentation, learning rate 0.5, 2000 epochs, batch size 64, cosine learning rate decay. The pretrained model and all the other hyper-parameters are defaulted from <https://github.com/HobbitLong/PyContrast>.

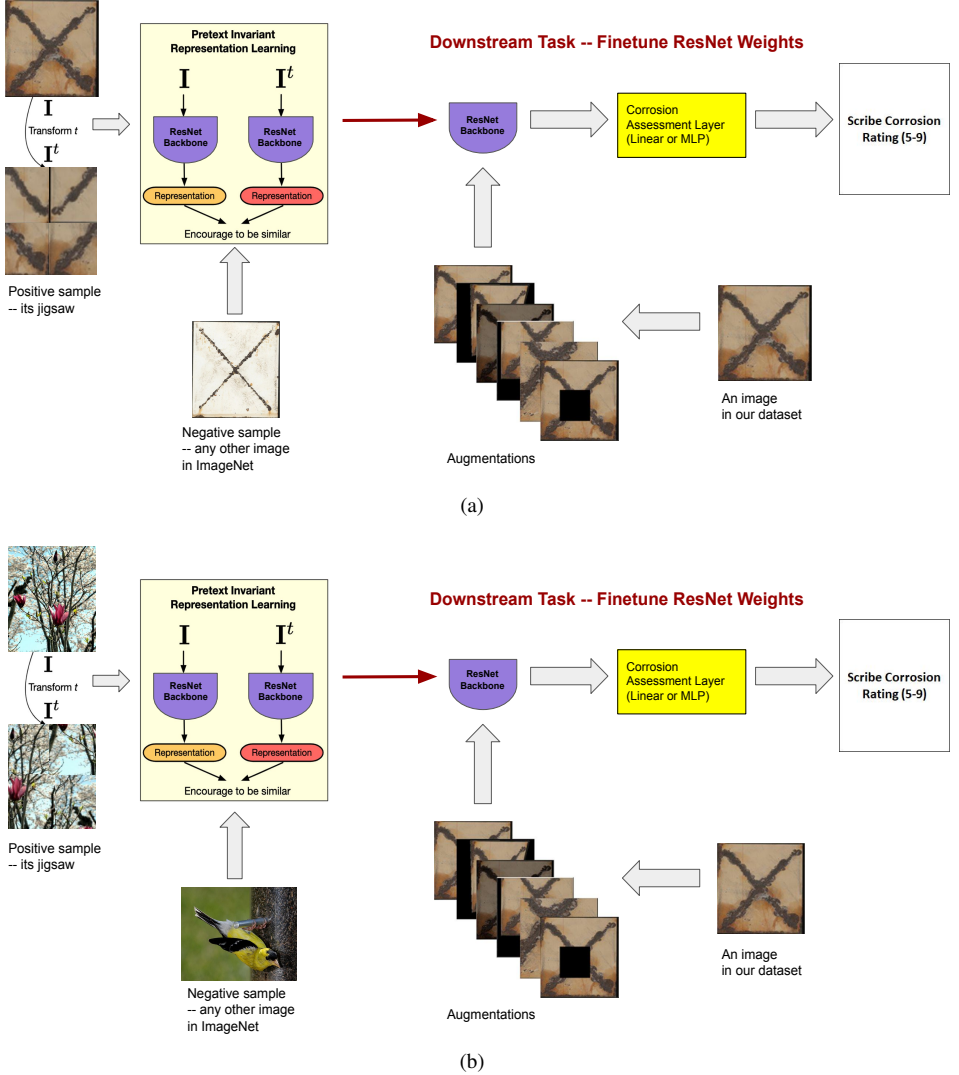


Figure 6: (a) PIRL from scratch to classify corrosion (b) Pretrained PIRL on ImageNet to classify corrosion

| Augmentation Method | Description | Parameters & Magnitude ranges searched |
|---------------------|--|--|
| Color Jitter | Randomly change the brightness, contrast, saturation, and hue of an image. Brightness, saturation, and contrast change is chosen uniformly from given (min, max) and should all be non-negative numbers. Hue is chosen uniformly from (-magnitude, +magnitude). | Brightness: (0.5, 1.5) (1.5, 2) Contrast: (0.5, 1.5) , (1.5, 2) Saturation: (0.5, 1.5) , (1.5, 2) Hue: 0.25, 0.5 Probability: 0.25 , 0.50, 0.75 |
| | | Kernel Size: 11 , 27 Sigma: 5 , 10 Probability: 0.25, 0.50, 0.75 Probability: 0.25 , 0.50, 0.75 Probability: 0.25, 0.50 , 0.75 |
| Horizontal Flip | Flip an image horizontally with some probability of likelihood. | Area Ratio: (0, 0.05) , (0.05, 0.15), (0.15, 0.5) Min Aspect Ratio: 0.3 Max Attempt: 1, 5 , 10, 20 Probability: 0.25 , 0.50, 0.75 |
| Vertical Flip | Flip an image vertically with some probability of likelihood. | |
| Random Erasing | Randomly erase areas of the image. Area ratio defines the range to uniformly sample from to determine the erasing box area, min aspect ratio defines the min value to determine the aspect ratio of the erasing box. Max attempt is how many times to sample possible aspect ratios and area ratios. | |
| | | |
| Random Perspective | Perform random perspective transformation with some probability. Distortion scale is how severe the degree to which the image is distorted, possible range of 0 to 1. | Distortion Scale: 0.25 , 0.35, 0.5 Probability: 0.25, 0.50, 0.75 |
| Random Rotation | Rotate an image. Degree range is used to select a degree value from that range and use. Expand is the option to expand the boundary of the image so corners of the rotated image are not cropped out. | Degrees: (-25, 25) , (-50, 50), (-75, 75) Expand: True Probability: 0.25, 0.50, 0.75 |
| Random Resized Crop | Crop an image to a random size. Scale is the scale range of the cropped image before resizing. Aspect ratio is the aspect ratio range of the cropped image before resizing. | Scale: (1/8, 0.3), (0.3, 0.7) Aspect Ratio: (1,1) Probability: 0.25 , 0.50, 0.75 |
| Random Crop | Crop an image at a random location. Padding is the pad value on each border of the image. Fill is the value to fill padded areas after cropping when using constant padding mode. Padding mode "constant" uses the fill value to pad the border, "edge" pads with the last value on the edge of the image, "reflect" pads with the reflection of the image while not repeating the last value on the edge, and "symmetric" pads with the reflection of the image while repeating the last value on the edge. | Padding: 2, 4 Fill: 0 Padding mode: constant , edge, reflect, symmetric Probability: 0.25, 0.50 , 0.75 |

Table 1: Descriptions of each method taken from PyTorch website: <https://pytorch.org/vision/stable/transforms.html>.

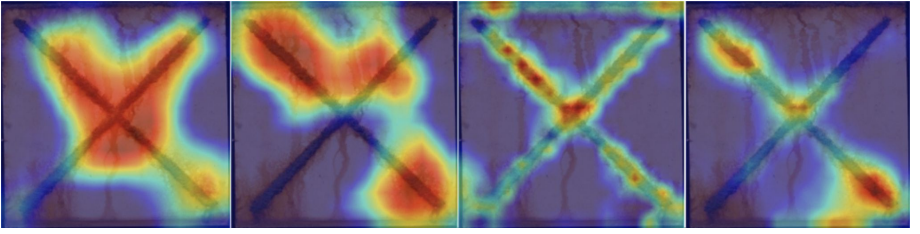


Figure 7: Additional Grad-CAM examples on a image with the ground truth corrosion rating 6. Left to right: ResNet-18, ResNet-50, DenseNet, HRNet. High activation in red, low activation in blue. All models predict this image as corrosion rating 5 is heavier than it should be.

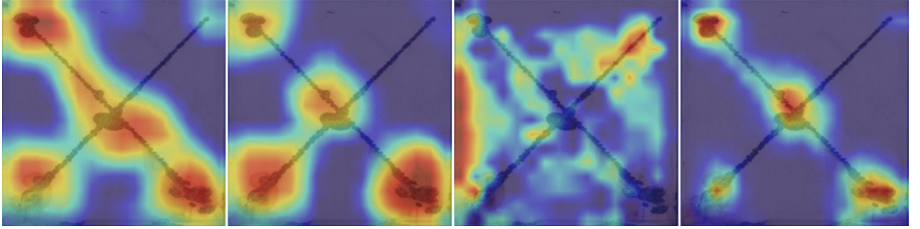


Figure 8: Additional Grad-CAM examples on a image with the ground truth corrosion rating 7. Left to right: ResNet-18, ResNet-50, DenseNet, HRNet. High activation in red, low activation in blue. All models predict this image as corrosion rating 5, which is heavier than it should be.

5 Grad-CAM

We add more Grad-CAM results to show the performance of our best trained models where they cannot predict our testing corrosion images correctly. In Figure 7, all models predict a sample test image, with the ground truth rating 6, as rating 5. In Figure 8, all models predict a sample test image, with the ground truth rating 7, as rating 5. From these results, we found CNNs may not be able to handle thin corrosion well. Instead, they focus on easily-learned areas, such as water staining or heavier corrosion.

6 Results and Conclusions

In supplementary Table 2 we see results presented in the main paper along with 2 additional results: pretrained supervised ResNet-50 with a MLP classifier and pretrained supervised ResNet-18 with a MLP classifier. These new model results are pretrained on ImageNet [9] and the MLP classifiers are trained for 2000 epochs using all the same hyperparameters used in the augmentation tuning experiments in Table 1 from the main paper. The pretrained networks are finetuned on our corrosion 10-fold training data sets, validated on the corresponding 10 validation data sets, and then evaluated on our test set of 60 images. We use the best combination of augmentation methods during the training process. For ResNet-18 model this includes a combination of random crop and color jitter and for ResNet-50 model this includes a combination of random crop, color jitter, random erasing, random perspective, and random resized crop.

In the top section of this table we see supervised results with no data augmentation to

| Method | Backbone | Classifier | Test Accuracy |
|--|-----------|------------|---------------|
| Supervised + No augmentation | ResNet-50 | N/A | 0.72 +/- 0.03 |
| Supervised + No augmentation | ResNet-18 | N/A | 0.78 +/- 0.03 |
| Supervised + Combo augmentation | ResNet-50 | N/A | 0.77 +/- 0.03 |
| Supervised + Combo augmentation | ResNet-18 | N/A | 0.81 +/- 0.04 |
| Supervised + Combo augmentation + Pretrained | ResNet-50 | MLP | 0.76 +/- 0.03 |
| Supervised + Combo augmentation + Pretrained | ResNet-18 | MLP | 0.81 +/- 0.03 |
| Supervised + Combo augmentation + Pretrained | ResNet-50 | Linear | 0.76 +/- 0.02 |
| Supervised + Combo augmentation + Pretrained | ResNet-18 | Linear | 0.83 +/- 0.01 |
| Self-Supervised + Pretrained | ResNet-50 | MLP | 0.75 +/- 0.03 |
| Self-Supervised + Pretrained | ResNet-50 | Linear | 0.68 +/- 0.02 |
| Self-Supervised | ResNet-50 | Linear | 0.72 +/- 0.04 |
| Self-Supervised | ResNet-18 | Linear | 0.70 +/- 0.03 |

Table 2: Results from main paper plus additional pretrained on ImageNet results using ResNet-18 and ResNet-50 with best combinations of augmentation methods found and presented in the main paper.

establish a baseline. The second part of this table contains the best supervised ResNet-50 and ResNet-18 results with the best combinations of augmentation methods; with these tuned augmentation methods we see that for both ResNet-50 and ResNet-18 we improve our test accuracy. In the third section, we see new results not in the main paper. These results are shown in order to demonstrate further that combining tuned data augmentation methods with pretrained models and a MLP classifier, we can further achieve similar performance to trained from scratch supervised and self-supervised performances. In the fourth section, we see the pretrained results from the main paper which use a linear classifier. Then, in the bottom section of the table, we again present our self-supervised representation learning results from the main paper. With all these results, we make several observations and list our findings here: i.) pretrained and supervised ResNet-18 with a linear classifier outperforms ResNet-18 trained from scratch when both are using tuned data augmentation methods, ii.) ResNet-50 trained from scratch outperforms pretrained and supervised ResNet-50 with a linear or MLP classifier also all with tuned data augmentations used, iii.) pretrained and supervised ResNet-18 with linear classifier outperforms pretrained and supervised ResNet-18 with a MLP classifier, iv.) pretrained and supervised ResNet-50 with linear or MLP classifiers perform the same, v.) pretrained and supervised ResNet-18 with linear classifier outperforms pretrained and supervised ResNet-50 with a MLP or linear classifier, vi.) pretrained and supervised ResNet-50 with linear classifier outperforms pretrained PIRL with ResNet-50 backbone and a linear classifier, vii) pretrained and supervised ResNet-50 with MLP classifier outperforms pretrained PIRL with ResNet-50 backbone and MLP classifier, and viii.) overall, pretrained and supervised ResNet-18 with the tuned data augmentations yield the best test classification performance (0.83).

References

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and*

- pattern recognition*, pages 770–778, 2016.
- [2] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
 - [3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
 - [4] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.
 - [5] ASTM Committee D-1 on Paint, Materials Related Coatings, and Applications. *Standard Test Method for Evaluation of Painted or Coated Specimens Subjected to Corrosive Environments*. ASTM International, 2008.
 - [6] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Minghui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.