CS230

# Semi-Supervised Classification of Images for Post-Earthquake Building Damage Assessment

Project Category: Computer Vision

**Aaron Appelle**
Stanford University
appellea@stanford.edu

**Link to code:** https://github.com/aaronappelle/CS230Project

## Abstract

Post-earthquake damage surveys currently require teams of domain experts to visually inspect buildings to determine their safety, which is slow and subjective. Efforts to automate the process using computer vision have been limited due to the time and resource cost of labeling earthquake survey images. In this project I use pseudo-labeling to take advantage of large numbers of unlabeled reconnaissance images available on the web in a semi-supervised learning approach. I investigate the effects of changing the start-epoch of pseudo-labeling during training. Results show that prediction accuracy improved by up to 5% on a test set from the unlabeled data, although improvement is sensitive to the accuracy of the base model.

## 1 Introduction

Following an earthquake, building safety evaluations must be done as quickly as possible to determine whether a building is safe for continued occupation [1]. The current practice for preliminary post-disaster evaluation is visual inspection [2], which is subjective to the reviewer [1, 3] and slow [4]. A computer vision framework to assess damage of structures from images is crucial for the eventual development of an autonomous reconnaissance system to replace manual inspection. Past work has demonstrated reasonably good results for supervised image classification tasks like collapse mode and damage state, with accuracy ranging from 60-90% [5]. However, these models require images to be manually labeled and sorted into independent tasks (Figure 1). Image labeling is an expensive and time-consuming job requiring teams of domain experts, leading to limited availability of labeled data. Real earthquake survey images are highly varied and uncurated with characteristics that may be different from existing training sets.

My contribution is a semi-supervised learning (SSL) approach to incorporate unlabeled survey images into the model's training, thereby providing a framework for future improvement of the model using new post-earthquake building assessment images. I tackle two separate classification tasks: Task 1 (Scene Level) and Task 2 (Damage State) (Figure 1) due to their public availability. Both classification tasks take as input images from earthquake reconnaissance surveys. The output are labels that help engineers categorize the photo to determine if the building would be safe for occupancy, described in more detail in Section 5.

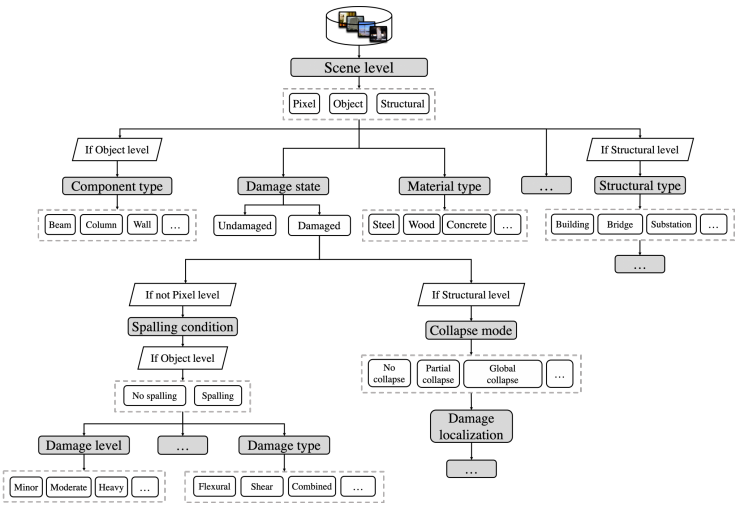


Figure 1: Hierarchy of multiple tasks for structural damage image classification on $\phi$-Net dataset [5]

## 2 Related work

**Building Damage Classification**: Past attempts to use computer vision for classification of earthquake reconnaissance images use a limited scope in order to simplify the classification problem. One type of scope limitation is on the type of damage, for example only detecting cracks on surfaces [6]. Another type of scope limitation is the type of infrastructure considered, for example detecting damage only on concrete bridges [7]. A third type of scope limitation is the nature of the input data [8] where the input image should be lab-quality to match the training data distribution. Gao and Mosalam [5] were the first to broaden the scope by publishing an open-source labeled image dataset ($\phi$-Net) for post-earthquake damage assessment containing variety and complex patterns of damage, scene, building types, etc. The most useful model should be able to handle images from any setting, including those not accounted for in the training dataset. There have not yet been any attempts to incorporate large amounts of unlabeled earthquake reconnaissance images into the training of classifiers.

**Semi-supervised learning (SSL)**: SSL serves as a training strategy to leverage unlabeled data to improve the model's performance, which is especially appropriate for this application given the large availability of unlabeled images and the high resource cost of labeling. Self-Training SSL methods [9, 10] use a model trained on labeled data to predict pseudo-labels for the unlabeled data. Those pseudo-labels are treated as ground truth in order to train the model simultaneously on labeled and unlabeled data. Consistency Regularization methods [11, 12] use input noise in training to generate predictions on unlabeled images that remain the same in the presence of noise. Hybrid methods such as Fix-Match [13] combine pseudo-labeling with consistency regularization using both weakly and strongly augmented images.

## 3 Dataset

I am using the PEER-Hub Image Net ($\phi$-Net) [5] published by Gao and Mosalam with the Pacific Earthquake Engineering Research center (PEER). The description of the image categories for each classification task is given in Table 1. A limitation of the dataset is that, although the makeup of the images for the two tasks is identical, the provided labels are single-attribute, i.e. Task 1 images are not labeled with Damage State, and Task 2 images are not labeled with Scene Level.[1] For *visual examples* of images included in the dataset, please see Figures 4 and 5 in the Appendix.

[1]The original authors of $\phi$-Net [5] are currently working to combine the datasets for multiple tasks into one multi-attribute dataset

| Classification Task | Label | Description |
|---|---|---|
| Task 1: Scene Level | Class 0: "Object" | Photo of a building component like column or wall |
| | Class 1: "Pixel" | Close-up image of a surface, ex. cracked wall |
| | Class 2: "Structural" | Photo containing entire building or multiple buildings |
| Task 2: Damage State | Class 0: "Damaged" | Visible damage (cracking, crumbling) on the building |
| | Class 1: "Undamaged" | No visible damage on the building or component |

Table 1: Description of the two $\phi$-Net classification tasks.

Task 1 (Scene Level) has 27,306 images in total with a roughly even spread between the three classes. There are 2997 images in the provided $\phi$-Net test set, which is approximately 10% of the dataset. I use a validation set size equal to 10% of the dataset, leaving 21,878 images for training. Task 2 (Damage State) has 13,271 images in total with a roughly even spread between the two classes. There are 1460 images in the provided $\phi$-Net test set, which is 11% of the dataset. I use a validation set size equal to 10% of the dataset, leaving 10,630 images for training.

For the SSL implementation, I manually gathered an additional dataset of unlabeled earthquake survey images from multiple sources. A first source was the the Civil and Environmental Engineering (CEE) department at Stanford from Prof. Eduardo Miranda[2], whose group regularly performs post-earthquake surveys for research after major events, ex. [14]. Secondly, I gathered thousands of images by webcrawling [15, 16]. The total size of the unlabeled image dataset is 8092 images. For evaluation of performance on an unlabeled test set, I manually labeled 200 images. The remainder of the unlabeled images (7892) are used for training by SSL. I did not manually label any images for inclusion in the validation dataset due to the time cost (which is reflective of real-life challenges). From the 200 labels, the test set appears to have a *class imbalanced* distribution: 91 "Object Level", 9 "Pixel Level", and 100 "Structure Level" for Task 1; 112 "Damaged", 88 "Undamaged" for Task 2.

In summary, I test the models on *two different test sets*:

1. **$\phi$-Net Test Set**: Task 1 and Task 2 test sets provided with $\phi$-Net, sized approx. 10% of the labeled training dataset (a few thousand images)
2. **Unlabeled Data Test Set**: A test set of 200 images from the unlabeled dataset which I manually labeled, whose distribution is different from the $\phi$-Net training data

All images are pre-processed by downsampling to 224x224 pixels with 3 channels (RGB). Images with low original resolution (below 448x448) or bizarre aspect ratios were manually discarded from the dataset. I use image augmentation techniques on the labeled training dataset by applying small amounts of rotation, translation, zoom, shear, rescaling, and horizontal reflection. None of the transformations distort the meaning of the images. Image augmentation is not used on the unlabeled image dataset.

## 4 Methods

For the baseline model, I use transfer learning with the architecture of VGG-16 [17] trained on ImageNet [18]. After discarding the pre-trained fully-connected layers of VGG-16, I add on 2D Global Max Pooling to reduce the image dimensions to one while retaining the number of channels, followed by two fully-connected (FC) layers to reduce the dimensionality to the number of output classes. The first FC layer has 512 hidden units and uses dropout with a keep probability of 0.5. The second FC layer has the same number of units as output classes (three for Task 1, two for Task 2). Then the models are trained separately for the two tasks. I chose to keep all pre-trained convolutional layers of VGG16 because high-level features in natural objects from ImageNet are greatly helpful in identifying features of buildings. Future work might explore the effect of re-training some of the later layers of VGG-16. Detailed metrics of the baseline models are in Table **??**.

For the SSL model, I use the Pseudo-Label Method [9]. In this method the model is trained simultaneously on labeled and unlabeled images. Cross-entropy loss is used for the labeled images, and pseudo-labels are generated for unlabeled images using the same model according to maximum confidence. The loss function is then a weighted sum of the loss of the labeled data and the loss of the unlabeled data: $L = L_{labeled} + \alpha(t) \cdot L_{unlabeled}$. Therefore $\alpha(t)$ determines the importance of

[2]https://profiles.stanford.edu/eduardo-miranda

the pseudo-labels over time. Standard practice is to start $\alpha$ at 0 at epoch $T_1$ and increase it linearly over time until some final training epoch $T_2$ [9]:

$$\alpha(t) = \begin{cases} 0 & t < T_1 \\ \frac{t-T_1}{T_2-T_1}]\alpha_f & T_1 \leq t < T_2 \\ \alpha_f & T_2 \leq t \end{cases}$$

Because the scheduling of $\alpha(t)$ is crucial for performance, I chose to run experiments varying $T_1$. If $T_1$ is too soon, it will disturb the training on labeled data. If $T_1$ is too late, then the benefits of using unlabeled data will diminish. I chose to fix $T_2$ to be the last training epoch, matching the training schedule of the baseline model. Otherwise, the model hyperparameters are the same as VGG-16, and the new layers are trained using the Adam optimizer with learning rate $1 \times 10^{-4}$ and batch size 32.

## 5 Results and Discussion

I am reporting the accuracy and F1 score on the two test sets described in Section 5. The F1 score is useful for evaluating the performance on the imbalanced test set from the unlabeled data. The $\phi$-Net test set is class-balanced, so accuracy is the most important. Additionally, I am reporting a measure of the model's prediction confidence, which is computed as the average of the softmax probability of the predicted class (across examples).

| | | **Task 1: Scene Level** | | | | **Task 2: Damage State** | | | |
|---|---|---|---|---|---|---|---|---|---|
| Test Set | Metric | *Baseline* | $T_1 = 5$ | $T_1 = 10$ | $T_1 = 15$ | *Baseline* | $T_1 = 5$ | $T_1 = 10$ | $T_1 = 15$ |
| $\phi$-Net | Accuracy | 0.9009 | **0.9046** | 0.9042 | 0.9006 | 0.8116 | 0.8062 | 0.8144 | **0.8171** |
| | F1 | 0.9014 | **0.9053** | 0.9040 | 0.9010 | 0.8116 | 0.8056 | 0.8143 | **0.8171** |
| | Confidence | **0.9466** | 0.9391 | 0.9427 | 0.9436 | **0.8838** | 0.8744 | 0.8768 | 0.8779 |
| Unlabeled | Accuracy | 0.7300 | **0.7850** | 0.7550 | 0.7650 | **0.7000** | 0.6800 | 0.6800 | 0.6950 |
| | F1 | 0.7250 | **0.7788** | 0.7466 | 0.7616 | **0.6986** | 0.6806 | 0.6806 | 0.6921 |
| | Confidence | 0.9513 | 0.9481 | **0.9633** | 0.9532 | **0.8952** | 0.8766 | 0.8756 | 0.8763 |

Table 2: Test Set Performance. All models except baseline are using SSL with the start epoch of pseudo-labeling indicated by $T_1$

**Task 1 (Scene Level)**

For Task 1, the baseline model using transfer learning achieves 90.6% accuracy on the provided $\phi$-Net test set, but only 73% accuracy on the unlabeled data test set. The discrepancy in performance largely comes from the fact that the unlabeled images come from a different image distribution than the $\phi$-Net training data. The SSL models do not significantly affect performance on the $\phi$-Net labeled test set, with all metrics being effectively the same as the baseline model.

SSL does noticeably improve performance on the unlabeled data test set, achieving 78.5% accuracy up from 73% in the baseline for the best SSL model with $T_1 = 5$. Comparing across the three SSL models, having an earlier start time to include unlabeled data led to the largest performance improvement. However, there is not a clear trend as the $T_1 = 10$ model performs worse than the $T_1 = 15$ model. Therefore we cannot conclude that starting the pseudo-labeling earlier is necessarily better. Nevertheless, all models which utilize the unlabeled training data outperform the baseline model.

When looking at the training history for the SSL models like in Figure 2, it is clear that there is a jump in training accuracy when the new training data is introduced at epoch 5. Also, the validaiton loss is no longer monotonic as the training depends on predictions of the model as part of the Pseudo-Label Method [9]. Setting $T_1 = 5$ may have been best in this instance becuase epoch 5 is approximately when the validation loss stopped decreasing in the baseline model (see Figure 7 in the Appendix).

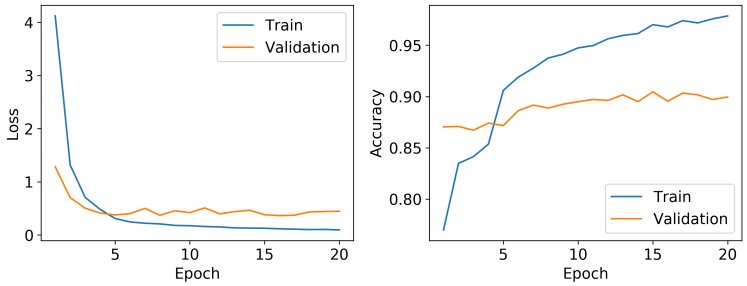


Figure 2: Task 1 training history for SSL with $T_1 = 5$

I checked image-by-image errors that the baseline model made on the unlabeled test set. The analysis reveals that the baseline model tends to struggle on indoor images which are classified as "Object Level" (Figure 3) due to being larger than a surface but smaller than a whole building. There is a scarcity of these types of photos in the $\phi$-Net dataset, explaining why SSL correctly labels them.

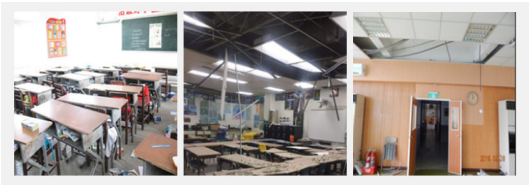

Figure 3: Images from the unlabeled test set that the baseline model mislabeled

**Task 2 (Damage State)**

For Task 2, the baseline model using transfer learning achieves 81.2% accuracy on the $\phi$-Net test set, and 70% accuracy on the unlabeled data test set. This is similar to the performance discrepancy observed in Task 1: again the images in the unlabeled test set are harder to classify when trained on the distribution of images in $\phi$-Net. For this classification task, SSL trivially improves classification performance on the $\phi$-Net test set but does not improve performance on the unlabeled test set.

It is useful to consider why SSL fails to improve performance on the unlabeled images for this task, seeing as it does improve performance in Task 1. One possible explanation is that the predictive performance of the model for Damage State is worse than for Scene Level (lower accuracy). The accuracy of predictions is crucial for the Pseudo-Labeling method, because it directly determines the proportion of correct pseudo-labels. In Task 2, it can be seen that the model is generally very confident on predictions (over 88%) but achieves relatively poor accuracy. Rizve et. al. [19] argue that Pseudo-Labeling "underperforms due to erroneous high confidence predictions from poorly calibrated models", and that the incorrect pseudo-labels lead to noisy training. This hypothesis is supported by my results. Given a highly accurate base model, the Pseudo-Labeling method should generally improve performance on test sets with different distributions as it can be considered equivalent to Entropy Regularization [9].

## 6 Conclusion/Future Work

Semi-supervised learning (SSL) using the Pseudo-Labeling (PL) method allows large quantities of unlabeled images to be used during training. For the task of classifying image Scene Level, PL using a set of unlabeled images successfully improved performance on a test set of images from that same set. For the task of classifying building Damage State, PL did not improve performance on the unlabeled test set. This discrepancy is likely due to the Damage State model having poor accuracy and therefore causing incorrect pseudo-labels. No strong trend emerged with the start epoch ($T_1$) of pseudo-labeling, although the results for Task 1 suggest that it would be a good idea to initiate PL at the epoch when validation loss begins to plateau. Future investigations can conduct sensitivity studies to the $T_1$ parameter, the accuracy of the supervised model, and the size of the unlabeled image dataset. Recent methods like FixMatch [13] would likely perform better than pseudo-labeling alone.

## 7 Contributions

All work is my own. The models were implemented using Keras [20] with TensorFlow backend [21]. Thank you to GitHub user koshian2 for the baseline Keras version of the Pseudo-Labeling method[3]. Thank you to Akhil Jhanwar for a guide on transfer learning with VGG-16. [4].

## References


[1] B. Galloway, J. Hare, D. Brunsdon, P. Wood, B. Lizundia, and M. Stannard, "Lessons from the post-earthquake evaluation of damaged buildings in christchurch," *Earthquake Spectra*, vol. 30, no. 1, pp. 451–474, 2014.

[2] Y. Reuland, P. Lestuzzi, and I. F. Smith, "A model-based data-interpretation framework for post-earthquake building assessment with scarce measurement data," *Soil Dynamics and Earthquake Engineering*, vol. 116, pp. 253 – 263, 2019.

[3] S. Lin, S. Uma, and A. King, "Empirical fragility curves for non-residential buildings from the 2010-2011 canterbury earthquake sequence," *Journal of Earthquake Engineering*, vol. 22, 12 2016.

[4] D. A. McEntire and J. Cope, *Damage Assessment After the Paso Robles, San Simeon, California, Earthquake: Lessons for Emergency Management*. Natural Hazards Center, 2004.

[5] Y. Gao and K. M. Mosalam, "Peer hub imagenet: A large-scale multiattribute benchmark data set of structural images," *Journal of Structural Engineering*, vol. 146, no. 10, p. 04020198, 2020.

[6] A. Zhang, K. C. P. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J. Q. Li, and C. Chen, "Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 805–819, 2017.

[7] X. Liang, "Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with bayesian optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 5, pp. 415–430, 2019.

[8] Y. Xu, S. Li, D. Zhang, Y. Jin, F. Zhang, N. Li, and H. Li, "Identification framework for cracks on a steel structure surface by a restricted boltzmann machines algorithm based on consumer-grade camera images," *Structural Control and Health Monitoring*, vol. 25, no. 2, p. e2075, 2018. e2075 STC-16-0276.R1.

[9] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, 2013.

[10] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," 2020.

[11] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," 2017.

[12] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," 2018.

[13] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[14] E. Miranda, S. Brzev, N. Bijelic, Z. Arbanas, M. Bartolac, V. Jagodnik, D. Lazarević, S. Mihalić Arbanas, S. Zlatovic, A. Acosta, J. Archbold, J. Bantis, J. Borozan, I. Božulić, N. Blagojevic, C. Cruz, H. Davalos, E. Fischer, S. Gunay, and I. Robertson, "Steer-eeri: Petrinja, croatia december 29, 2020, mw 6.4 earthquake joint reconnaissance report (jrr)," 01 2021.

[15] StEER (Year) NSF Structural Extreme Events Reconnaissance (StEER) Network, "Official steer products," *Fulcrum Community*.

[16] Earthquake Engineering Research Institute, "Learning from earthquakes reconnaissance archive."

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[19] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," 2021.

[20] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[21] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.


[3] https://github.com/koshian2/Pseudo-Label-Keras

[4] https://medium.com/analytics-vidhya/cnn-transfer-learning-with-vgg16-using-keras-b0226c0805bd

# Appendix

## Data


Figure 4: Example of Task 1 images labeled with Scene Level in $\phi$-Net [5]

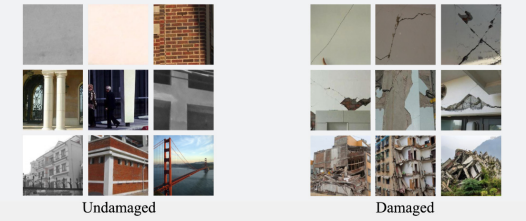


Figure 5: Example of Task 2 images labeled with Damage State in $\phi$-Net [5]

## Task 1 Results

Model Training Histories:

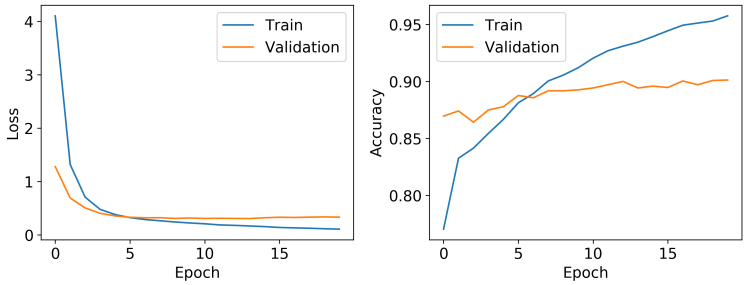


Figure 6: Task 1 training history for baseline supervised model

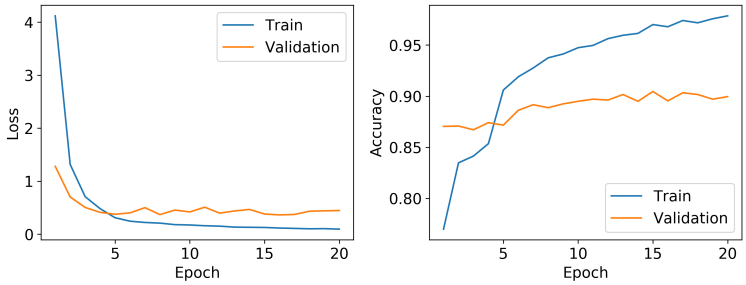


Figure 7: Task 1 training history for best SSL model, $T_1 = 5$

Performance on $\phi$-Net test set:

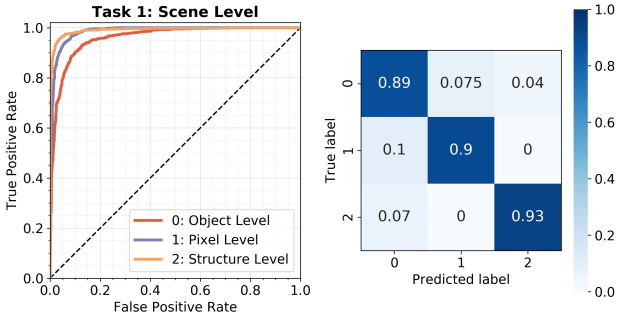


Figure 8: Baseline Task 1 Performance on $\phi$-Net test set

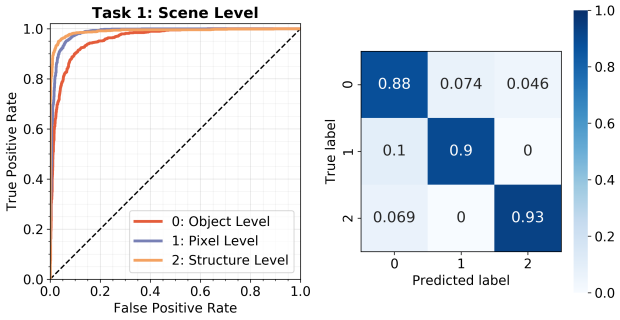


Figure 9: Best SSL Model ($T_1 = 5$) Task 1 performance on $\phi$-Net test set

Performance on unlabeled data test set:

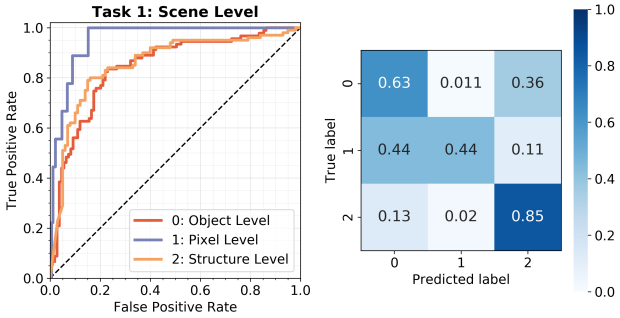


Figure 10: Baseline Task 1 performance on unlabeled data test set

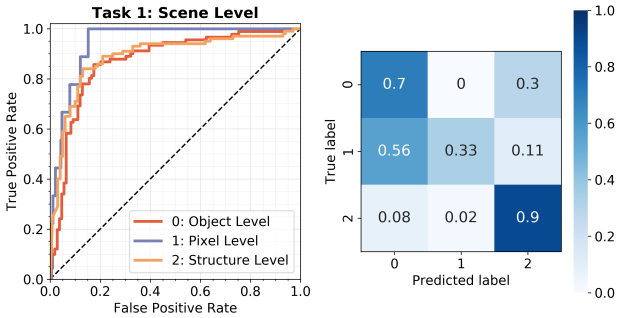


Figure 11: Best SSL Model ($T_1 = 5$) Task 1 Performance on unlabeled data test set

**Task 2 Results**

Model Training Histories:

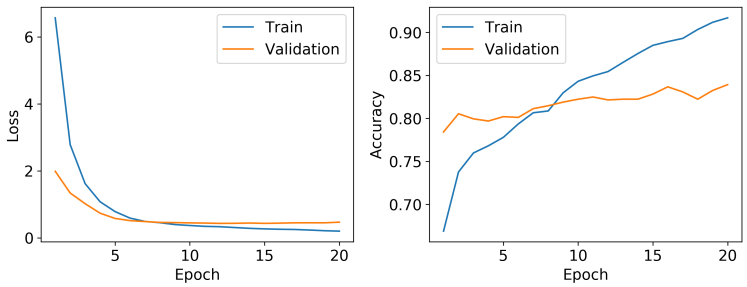


Figure 12: Task 2 training history for baseline supervised model

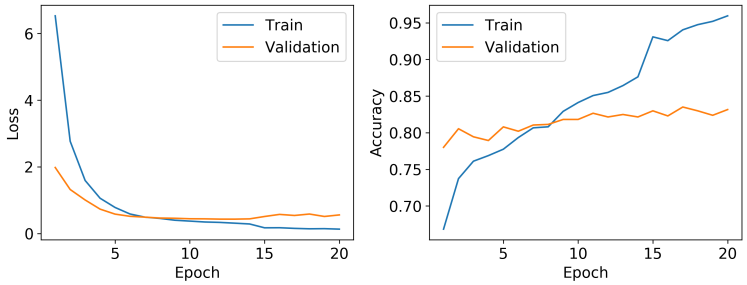


Figure 13: Task 2 training history for for best SSL model, $T_1 = 15$

Performance on $\phi$-Net test set:

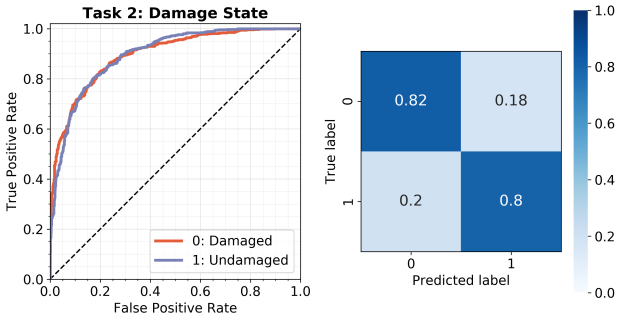


Figure 14: Baseline Task 2 Performance on $\phi$-Net test set

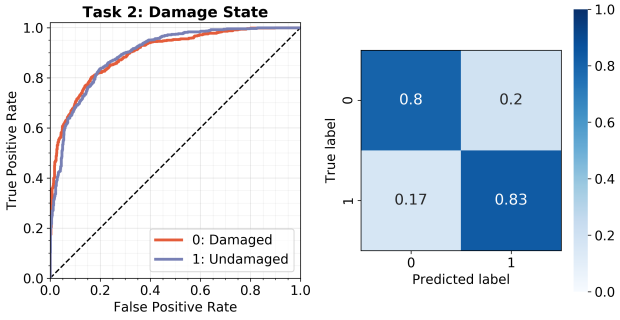


Figure 15: Best SSL Model ($T_1 = 15$) Task 2 performance on $\phi$-Net test set

Performance on unlabeled data test set:

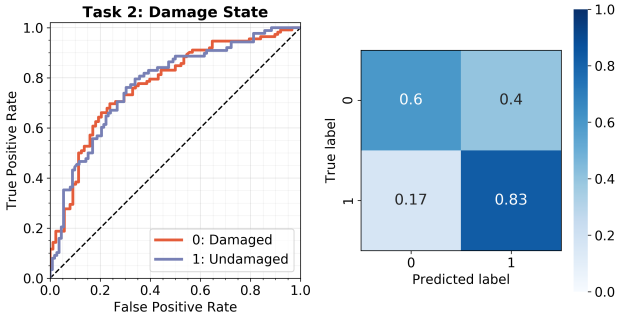


Figure 16: Baseline Task 2 performance on unlabeled data test set

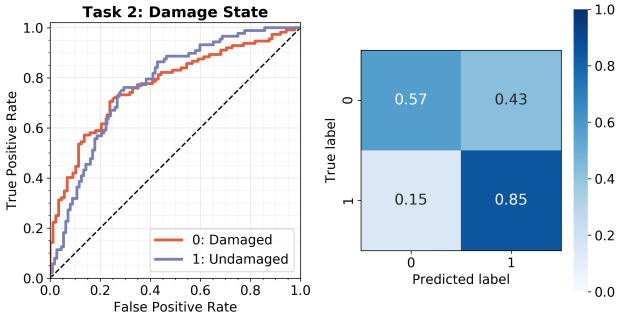


Figure 17: Best SSL Model ($T_1 = 15$) Task 2 Performance on unlabeled data test set