TRANSFER LEARNING FOR BUILDING DAMAGE ASSESSMENT *

Yandi Wu
University of Wisconsin, Madison
Madison, WI, USA
yandi.wu@wisc.edu

Charlotte Ellison
U.S. Army Corps of Engineers
Engineer Research and Development Center
Geospatial Research Laboratory
Alexandria, VA, USA
Charlotte.L.Ellison@usace.army.mil

ABSTRACT

After a natural disaster or battle, damage assessment can be costly, time-consuming, and, in the case of on-the-ground damage detection, a potentially dangerous process. An alternative to on-site or manual battle damage assessment is image segmentation of satellite imagery taken after the disaster or battle, namely, identifying and labeling buildings by damage level. Image segmentation can be achieved by training a convolutional neural network when imagery is readily available. However, battle damage data is limited, difficult to create, and typically focused in a single geographic region. Because of this, one may want to repurpose a pretrained model to perform damage assessment on a test set drawn from a different distribution as the training set. Unfortunately, a neural network trained to perform image segmentation of images from one location may perform poorly on a test set from a different location. Two specific transfer learning techniques, stochastic weight averaging (SWA) and multidomain adaptive batch normalization (multiadaBN) are designed to address this out-of-domain generalization problem, and will be the focus of this paper. We find that in the case where the training set is small, as is often the case in the real world, the effectiveness of SWA and multiadaBN depends on both the type of training data and training set, and in some cases, the two methods even decrease model performance.

***K*eywords** Domain Adaptation · Semantic Segmentation · Building Damage Detection

1 Introduction

Assessing battle damage is a crucial step in the Joint Targeting Cycle (CJCSI 3162.02) and for general situational awareness in areas of conflict. Battle Damage Assessment (BDA) is often performed under less-than-ideal conditions due to limited data and time, which makes it ripe for automation using neural networks. However, one major downside of neural networks is that they are data-greedy and in an ever-shifting battle space or global geopolitical environment, acquiring labeled training data from an area of interest may not be feasible. Hence, this work focuses on using transfer learning techniques to take models pretrained over a small, previously labeled dataset and apply them to new areas.

There are several ways to collect data for building damage detection (BDD), one aspect of BDA. One method is a field survey, which requires onsite data collection. Field surveys are costly, time-consuming, and unviable if the target site is unsafe to visit. A safer option is collecting data via remote sensing, such as satellite or drone imagery. While one can employ specially trained analysts to label remotely sensed data, manual data labeling is often difficult, time-consuming, and prone to human error. Automated damage assessment can provide a cheaper, faster, and potentially more accurate solution to the aforementioned shortcomings of manual BDA.

Convolutional neural networks (CNNs) have been shown to effectively label satellite imagery from natural disasters when the test and training data are drawn from the same distribution (identically, independently distributed, or IID). However, this is an unrealistic scenario. A better alternative may be to repurpose a CNN trained on previous disasters or battles to perform BDD on a new test set from a different location that had a limited size of collected data. This is the

problem of Domain adaptation (DA) applied on satellite imagery, which are still in their early stages of development, and two of which we explore two of these techniques in this paper.

In this paper, we first give an overview of the problem at hand, and previous work done to address the problem. We then introduce the dataset, xBD, and our choice of training and test data designed to mimic battle data. Next, we introduce the two transfer learning techniques we will be testing, SWA and multiadaBN, and our results from using various training and testing data. Finally, we suggest some future work for BDD.

2 Related Works

In 2019, the Defense Innovation Unit launched the xView2 challenge, which invited participants to design a neural network to perform semantic segmentation on satellite images taken before and after disasters (Gupta et al; https://xview2.org/). There were two main tasks in the xBD challenge: building localization, the detection of buildings in the image, and damage assessment, the assignment of a damage level to each pixel. Gupta et al also introduced a baseline UNet backbone model with a low F1 score. See the dataset section for more details. The challenge produced some well-performing results, with the top performers achieving an F1 score of over 80 percent.

Gupta and Shah, however, observed that their model did not perform as well when the test data was drawn from a different set of disasters as the training data (out of domain, or OOD setting). They proposed a Binary Cross Entropy loss function in response to the OOD generalization problem and observe moderate improvements in results.

Outside the realm of data assessment via remotely sensed imagery, OOD generalization, or DA, is a well-studied problem. PixelDA, the prototypical pixel-level DA model, is based on generative adversarial networks (GANs) and alters images from the source domain to resemble those from the target domain (Bousmalis et al). Feature-level DA methods transform the source and/or target domains to minimize the discrepancies between the two via feature extraction. An example is Generate to Adapt, proposed by Sankaranarayanan et al, which brings the target and source domains closer together in a joint feature space using a GAN. Tzeng et al propose Adversarial Discriminative Domain Adaptation (ADDA), which also matches the two domains by extracting features from the target domain to align with the source domain until a discriminator cannot distinguish between the target and source. Hybrid methods have also been proposed, such as Contrastive Adaptation Network (CAN), which alternates between clustering and adaptation by minimizing intra-class discrepancies and maximizing inter-class margins (Kang et al).

Despite such advances, there is not yet much literature on image segmentation of satellite images in the OOD setting. Benson and Ecker apply stochastic weight averaging (SWA) and multidomain batch normalization (multiadaBN), two transfer learning techniques, to two models: one with a ResNet50 backbone and the other with a HRNet backbone. They observe improvements in the F1 scores when both methods are applied in the OOD setting, with the results more noticeable for the ResNet50 model. Since Benson and Ecker's code is publicly available and well-documented, their work serves as the basis for this paper, although other work on transfer learning on satellite imagery has been done. Lin et al propose a pair of GANs to perform image segmentation on OOD hurricane data. For instance, they use data collected from Hurricane Sandy to train their GANs and Hurricane Irma data for testing. They report a F1 score of at least 81 percent in all scenarios and show their GANs outperform other models adapted for transfer learning. Valetjin et al (2020) use a model with an Inceptionv3 backbone and divide their test and training data by disaster type, in particular focusing on wind and water disasters. They found using the Joplin tornado for the OOD testing (and other wind disasters for training) yielded much better results than using the Nepal flood for testing (and other water disasters for training). In a way, some of their results mirror ours, showing it can be hard to predict what factors influence transferability.

3 Dataset

We work with the xBD dataset, as it contains examples of damage from across the globe and spans different disaster types, providing a varied testbed on which to explore the effect of transfer learning on OOD performance. It consists of 22,068 pre- and post-disaster RGB satellite images of size 1024 x 1024 pixels introduced as part of the xView2 challenge (Gupta et al 2019). The dataset also provides polygon-labeled buildings for the 10,101 pairs of satellite images that are part of its training and validation sets. Each building is also labeled with "no damage," "minor damage," "major damage," and "destroyed" (see Figure 1). The remaining pairs of images are set aside in the "hold" set, which was not made publicly available until after the xView2 challenge and used by the organizers to evaluate model performance. We use a subset of the "hold" data for IID model performance evaluation.

We also consider the "Tier 3" dataset that was later added to the original xBD dataset, which contains disasters not present in the original dataset associated with the challenge and thus serves as the perfect candidate with which we can study OOD generalization. See the experiments section for more details.

Table 1: Pixel counts of two datasets of training data

	No Damage	Minor Damage	Major Damage	Destroyed	Total
Most Damaged Dataset	5842894 (53.07%)	200768 (1.82%)	193245 (1.76%)	4773680 (43.36%)	11010587
Balanced Dataset	10053437 (67.4%)	270168 (1.81%)	217187 (1.46%)	4368461 (29.3%)	14543366

Table 2: Pixel counts of two datasets of validation data

	No Damage	Minor Damage	Major Damage	Destroyed	Total
Most Damaged Dataset	3246976 (71.42%)	86317 (1.90%)	25081 (0.55%)	1188238 (26.13%)	4546612
Balanced Dataset	4235552 (86.29%)	114914 (2.34%)	43771 (2.34%)	514039 (10.47%)	4908276

4 Methods

Since our main objective is battle damage assessment, we focused on disasters whose post-disaster images resemble a war zone. This rules out, for example, disasters such as flooding and volcano eruptions. In the end, our training set consisted of images from the Mexico City earthquake and Santa Rosa and SoCal fires, as fire damage is often seen in a war zone, and densely populated cities like Mexico City are commonly targeted.

The xBD data contains high amounts of negative imagery, or images with little to no damaged buildings, often collected from rural areas. Data collected over a warzone, however, would not exhibit high levels of negative imagery, as the area of interest is likely to be known and specific. To reflect a realistic BDA dataset, data is limited to a small training set with no negative imagery. Testing data is also restricted to just fire damage, but with the realization that one could benefit from analyzing data from more urban settings. We now explain our training and test data selection, as well as other details about our experiments, such as the model, DA techniques, and scoring techniques used.

4.1 Training Data

We now explain how to create training data similar to the smaller urban dataset with high levels of damage that we might collect after a battle. We use two sets of training data, both containing 150 images. The first dataset, which we call the most damaged dataset, consists of images with the highest number of pixels labeled as damaged. As demonstrated by Table 1, most of these pixels are labeled as "destroyed." The second dataset consists of a 50/50/50 split of images with the highest number of pixels labeled as "minor damage," "major damage," and "destroyed" respectively. This balanced dataset contains more examples of minor and major damage, the rarest damage categorizations. Images that appear in the top 50 in more than one list are replaced with another image further down the list. In the end, the two datasets had 108 images in common, which is not surprising since images with high levels of damage are likely to be chosen for both training sets.

The following table shows the number of pixels in each damage category. Since there are not many examples of major damage in the training dataset, for the balanced dataset, all of the samples exhibiting any major damage are used. Some of the images with major damage tended to have more negative imagery, so the proportion of damaged pixels labeled as major and minor damage is similar for the two datasets.

A similar technique was used for validation. We selected 35 images for two separate validation datasets. The first "most damaged" dataset contains 35 images from the original validation set for the xView2 challenge with the most damaged pixels. The "balanced" dataset contains a 12/12/11 split of images with the most pixels labeled by "minor damage," "major damage," and "destroyed" respectively. The two validation datasets shared 22 images, but the proportions of damage levels are different (see Table 2). In particular, the balanced dataset contains a higher proportion of negative imagery and more pixels labeled with minor and major damage while the most damaged dataset contains a large number of destroyed pixels and more damaged pixels.

4.2 Model

We used the Dual HRNet model developed by Koo et al, which placed fifth in the xView2 challenge. Dual-HRNet consists of two ImageNet pretrained HRNet backbones developed by Wang et al. One is used for localization and

Table 3: Classification data for testing and training sets (pixel count)

	Background	No Damage	Minor Damage	Major Damage	Destroyed
Tornado	12630400 (81.55%)	355299 (2.29%)	568030 (3.67%)	452979 (2.92%)	1481292 (9.56%)
Water	13658133 (88.19%)	803703 (5.19%)	347902 (2.25%)	441801 (2.85%)	236461 (1.52%)
Fire	14936760 (96.44%)	289448 (1.87%)	20724 (0.13%)	29755 (0.19%)	211313 (1.36%)
Train	146275813 (93.00%)	5842894 (3.71%)	200768 (1.28%)	193245 (1.23%)	4773680 (3.04%)

one for classification, amalgamated by three fusion blocks. To address the class imbalance problem, the authors use the Lovasz softmax function. We chose this model because it is a high-ranking entry that only uses one model for its backbone, as opposed to the winning entry, which is an ensemble of multiple networks and would thus be hard to train with limited GPU memory.

In accordance with the original training scheme, the input data consists of random cropping of the original 1024 x 1024 pixel sample images into 512 x 512 pixel tiles, followed by standard augmentation techniques. The model is trained with a Nvidia RTX 2060 GPU over 500 epochs with a SGD optimizer with a base learning rate of 0.01. Since we had very limited GPU memory, we decrease the batch size to 1 (compared to 32 in Koo et al and 6 in Benson and Ecker).

4.3 Methods for improving OOD Generalization

Benson and Ecker employ two methods for OOD generalization: stochastic weight averaging (SWA) and multi-domain batch normalization (multiadaBN).

There are two main ways in which a model trained with SWA differs from a typical model (Izmailov et al). First, SWA uses a SGD optimization function with a constant or cyclical learning rate; in our case, we found a constant learning rate of 0.001 was optimal, after running various experiments. Secondly, SWA stores the weight parameters of the last specified number of epochs and averages them in the end for the final model. In our case, following Benson and Ecker, we store the weight parameters of 40 epochs after training for 500 epochs. SWA is a popular method for OOD generalization since its publicly available code is not hard to implement and has relatively little computational overhead.

Batch normalization (BatchNorm) is a standard way to stabilize and speed up neural networks, but the BatchNorm layers use the distribution of the training sample, which can differ greatly from the distribution of the test sample. Adaptive batch normalization (adaBN) addresses OOD generalization by recalculating the BatchNorm layers during testing using statistics from the test sample (Schneider et al). Since there are multiple disasters in the training set of our data, we are training across multiple domains rather than a single domain. As a result, the original adaBN is not suitable for training in multiple domains, so Benson and Ecker develop multi-domain adaBN (multiadaBN), which trains each disaster in separate mini-batches instead of using a mixture distribution that includes all the domains.

4.4 Testing

Following the lead of Gupta and Shah, we use the Tier 3 dataset for OOD testing. As with the training data, we exclusively use datasets exhibiting fire damage to reflect our end goal of BDA. In the end, we drew data only from the Portugal wildfire, Pinery bushfire, and Woolsey fire disasters.

Due to our very limited GPU memory, we could not run testing on some of the models without cropping the input image. After running experiments, we determined that 880 x 880 pixels was the maximum size of the input image we could use. As a result, we applied center crop to all our test images, making sure to also crop the ground truth data.

We also explored how well the models generalize to other disasters. We chose two other sets of disasters: tornados and water disasters from both the Tier 3 and original holdout data. Each test set, like the original fire damage test set, contained a sample of 20 images exhibiting the most damage. The tornados dataset drew samples from the Joplin, Tuscaloosa, and Moore tornados while the water dataset drew samples from the Midwest and Nepal floods and the Palu tsunami (although none of the Midwest flood data was ultimately selected). The distribution of pixels for each test set is shown in Table 3. Note that compared to the other test sets, the original fire damage dataset had the highest percentage of background pixels, meaning it tended to be the least urban. We tested the model trained with the most damaged dataset, since the test set consisted of samples with the most damage.

Table 4: MultiadaBN F1 scores for Tier 3 fire vs water disasters

	total F1	Damage F1	Loc F1	No Damage	Minor Damage	Major Damage	Destroyed
Tier 3 Fires	xView2 metric: 0.1832 Averaged F1: **0.4787**	xView2 metric: 4.0x10-6 Averaged F1: **0.4221**	0.6106	**0.7050**	0.03158	0	**0.5300**
Tier 3 Water	xView2 metric: **0.1931** Averaged F1: 0.3610	xView2 metric: 4.0x10-6 Averaged F1: 0.2399	**0.6437**	0.5222	**0.03833**	0	0.1591

4.5 Scoring

The original xView2 challenge scoring calculates F1total = 0.3F1loc + 0.7F1dmg, the weighted average of the localization and damage F1 scores.

The damage F1 scores is calculated by:

$$F1_{dmg} = \frac{4}{[(F1_{nodmg} + \epsilon)^{-1} + (F1_{mindmg} + \epsilon)^{-1} + (F1_{majdmg} + \epsilon)^{-1} + (F1_{dest} + \epsilon)^{-1}]} \tag{1}$$

The subscripts of the F1 scores in the denominator denote the damage levels, and EPSILON = 10-6 to prevent division by 0. This damage F1 score is challenging to work with because it very heavily penalizes a low F1 score in any category. In other words, the lowest damage F1 score will dictate the order of magnitude of the total damage F1 score. This is not a suitable metric for differentiating instances where one damage category has very low F1 scores, which is true in our setting with a small training dataset size. For example, consider Table 4, which consists of F1 scores of the model trained with the most damaged dataset and multiadaBN and tested on Tier 3 fire and water disaster data. As expected, since the model was trained on fire data, it does a much better job classifying destroyed buildings in the fire test set, as well as buildings with no damage. There is a marginal difference in its ability to classify minor damage. Despite an overall superior performance on the fire disaster test data, since the F1 scores for major damage are 0 for both datasets, the two datasets have the same damage F1, which is about 4 EPSILON.

An alternative way to calculate the damage F1 is to take the average of the F1 scores in each individual category. As seen in the table, the low minor and major damage F1 scores still heavily penalize the damage F1 score, but it is now more apparent that the model performed better on the Tier 3 fire data. Since the F1 score for major and minor damage is generally very low in our setting with a small training set, we will be calculating damage F1 by averaging the F1 scores.

5 Results

We now state the main conclusions we derived from our experiments.

5.1 The efficacy of multiadaBN and SWA depends on the disaster type

Unlike what we see with the fire and water damage data, multiadaBN and SWA appear to improve the tornado damage F1 scores, with little to no effect on the localization scores for all three disaster categories (see Table 5). For the water data, the baseline exhibits the best F1 scores with the exception of the destroyed F1 score, which multiadaBN and SWA both appear to improve. On the other hand, for the tornado data, the model with both multiadaBN and SWA exhibits the best F1 scores with the exception of the major damage F1 score. For the fire data, there is no clear conclusion for which model performs best, but the damage F1 scores appear to decrease with the application of multiadaBN. Upon examination of the confusion matrix, we find multiadaBN increases the proportion of buildings misclassified as destroyed (recall). See Figure 4 for examples.

5.2 The efficacy of multiadaBN and SWA depends on the training set

MultiadaBN and SWA appear to improve the model trained on the balanced training set, but not the most damaged training set. Table 5 (data for models trained with the most damaged training set) indicates that applying multiadaBN decreases the damage F1 score while applying SWA does not significantly affect it. Neither method appears to affect the localization F1 score significantly. On the other hand, applying both methods appears to improve the localization F1 score for the balanced dataset, and applying the two methods in conjunction improves the damage F1 score as well.

Table 5: Summary of F1 scores

	Baseline	SWA	multiadaBN	multiadaBN + SWA
Most damage fire	0.4461	**0.4589**	0.4049	0.4136
Balanced fire	0.3565	0.3804	0.3648	**0.4131**
Most damage tornado	0.4086	0.4353	0.4413	**0.4610**
Most damage water	**0.3288**	0.3109	0.319	0.3274

Table 6: Generalization gaps

	Baseline	SWA	multiadaBN	multiadaBN + SWA
Most damage fire	0.0923	**0.0703**	0.1485	0.1186
Balanced fire	0.1298	0.0939	0.1121	0**.0712**
Most damage tornado	0.1531	0.1675	0.1810	**0.1299**
Most damage water	0.1254	0.0980	0.1352	**0.0442**

5.3 MultiadaBN universally decreases the model's ability to classify minor and major damage

All models struggle to classify major and minor damage, regardless of the training or testing set used. MultiadaBN appears to adversely affect the models' ability to classify these damage categories, consistently decreasing the F1 minor damage scores and failing to detect any major damage. Even with the IID test data, there is no major damage detected once we apply multiadaBN; this suggests multiadaBN is not effective for improving F1 scores of categories that perform poorly when the baseline is tested on IID data.

5.4 The amount of negative imagery appears to affect localization scores

When tested with the model with the most damage, the tornado, water, and fire test sets had average localization F1 scores of 0.7377, 0.6326, and 0.6096 respectively. The tornado dataset, which exhibited the highest building to background ratio (see Table 3), had the highest localization score while the fire data, which exhibited the lowest building to background ratio, scored the lowest. The water test set had a higher background to building ratio than the fire dataset but only performed slightly better. We hypothesize this is because the tornado dataset consists of relatively evenly distributed buildings while the water disaster dataset consists of unevenly and densely distributed buildings; the former is easier to localize by hand. In addition, the tornados all happened in the US, like the majority of the samples in the training dataset, while the water disasters occurred in Asia and the fire disasters in Europe or Australia, where the terrain and building architecture are different.

We see a drop in localization score when we use the model trained with the balanced dataset on the fire test data. Compared to the more damaged dataset, a higher percentage of the more balanced dataset consists of buildings. With the more balanced dataset, we see a much higher percentage of buildings being misclassified as background (recall). This is because a model trained on a dataset with more buildings may have a harder time detecting small and sparsely distributed buildings, like those in the fire test data.

5.5 The generalization gaps give a different perspective on whether the transfer learning techniques were effective

Given a specific model, the generalization gap is the difference between the models' IID and OOD F1 scores. From this perspective, the model with both multiadaBN and SWA performed better than the other models in all but one scenario (most damaged training set, fire test set), in which the model with SWA performed the best. We also observe that in all but one scenario, SWA decreases the generalization gap and in all but one scenario, multiadaBN increases it. See Table 6 for details.

6 Conclusions and Future Work

Due to discrepancies in performance across the disaster types and training sets, we are unable to form a conclusive statement about the overall efficacy of multiadaBN or SWA. While multiadaBN and SWA appeared to improve F1 scores for the tornado data, the baseline performed the best for the water damage data while the SWA only model performed best for the fire damage test data. A small training set appears to hinder classification of major and minor damage, a problem that multiadaBN exacerbates. In fact, none of the models with multiadaBN applied were able to

classify major damage. On the other hand, the generalization gaps, which provide a different perspective of model performance compared to total F1 score, suggest the model with both SWA and multiadaBN outperforms other models in most scenarios tested. Exploring these transfer learning techniques on various subsets of the xView2 dataset has provided us with lessons we can use while creating tools for BDA, a scenario in which training datasets are scarce.

In order to improve our analysis, the data imbalance problem should be addressed: since buildings with minor and major damage are underrepresented in the training data, all models struggle to classify them. The data imbalance problem is common with smaller datasets one might be more likely to work with in the real world. The fact that multiadaBN decreases a model's ability to classify major or minor data may depend on the fact that major and minor damage F1 scores are universally low due to data imbalance.

Transfer learning applied to satellite imagery is not yet common, despite the humanitarian benefits of developing such techniques. On the other hand, there is abundant literature on applying transfer learning techniques for image segmentation of cityscapes using synthetic datasets, which are much easier to label. For example, the VisDA challenge in 2017 had an image segmentation track, which asked participants to use a model pretrained with the Grand Theft Auto 5 (GTAV) dataset to perform image segmentation on dashcam data (https://ai.bu.edu/visda-2017/). Drawing inspiration from using synthetic data from GTAV to segment real world data, one could use satellite imagery from the GTAV video game to localize satellite imagery, and possibly even classify damage, since fire damage is common in GTAV. Alternatively, one could adapt the transfer learning methods developed by participants of the VisDA challenge to create models for satellite imagery.

Lastly, although there exist transfer learning surveys for image classification (Zhuang et al), a comprehensive transfer learning survey for semantic segmentation would be helpful due to the large volume of literature available specifically for the GTAV/Cityscapes transfer learning problem. A comparison of the performance of various models, and analysis of which transfer learning methods are helpful in a certain scenario, would especially be helpful.

Acknowledgments

This research was supported in part by an appointment with the National Science Foundation (NSF) Mathematical Sciences Graduate Internship (MSGI) Program sponsored by the NSF Division of Mathematical Sciences. This program is administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and NSF. ORISE is managed for DOE by ORAU. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of NSF, ORAU/ORISE, or DOE.

The use of trade, product, or firm names in this document is for descriptive purposes only and does not imply endorsement by the U.S. Government. The findings of this document are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

References

[1] Benson, V. Ecker, A (2020). Assessing out-of-domain generalization for robust building damage detection. NeurIPS 2020 Workshop on Artificial Intelligence for Humanitarian Assistance and Disaster Response. https://arxiv.org/pdf/2011.10328.pdf

[2] Bousmalis K., Silberman N., Dohan D., Erhan D., and Krishnan D. Unsupervised pixel-level domain adaptation with generative adversarial networks. Proc. IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Jul. 2017, pp. 3722–3731.

[3] Gupta, R., Hosfelt, R., Sajeev, S., Patel, N., Goodman, B., Doshi, J., Heim, E., Choset, H., Gaston, M. (2019). xBD: A dataset for assessing building damage from satellite imagery. arXiv. https://arxiv.org/abs/1911.09296

[4] Gupta, R. and Shah, R.. Rescuenet: Joint building segmentation and damage assessment from satellite imagery. 25th International Conference on Pattern Recognition (ICPR2020), IEEE, Jan. 2020, pp. 4404-4411.

[5] Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G. Averaging weights leads to wider optima and better generalization. arXiv:1803.05407, 2018. URL https://arxiv.org/abs/1803.05407

[6] Kang, G., Jiang, L., Yang, Y. and Hauptmann, A.G. Contrastive adaptation network for unsupervised domain adaptation. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, Jun. 2019, pp. 4893–4902.

[7] Koo, J., Seo, J., Yoon, K., Jeon, T. (2020). Dual-HRNet for building localization and damage classification. Github. https://github.com/DIUx-xView/xView2_fifth_place/blob/master/figures/xView2_White_Paper_SI_Analytics.pdf

[8] Saenko, K. "Visual Domain Adaptation Challenge." Boston U., 2017, https://ai.bu.edu/visda-2017/. Accessed 19 August 2022.

[9] Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R. Generate to adapt: Aligning domains using generative adversarial networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, 8503-8512.

[10] Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., Bethge, M. Improving robustness against common corruptions by covariate shift adaptation. Advances in Neural Information Processing Systems, 33, 2020. URL https://proceedings.neurips.cc/paper/2020/file/85690f81aadc1749175c187784afc9ee-Paper.pdf Tzeng E., Hoffman J., Zhang N., Saenko K., Darrell T. (2014). Deep domain confusion: Maximizing for domain invariance. arXiv. http://arxiv.org/abs/1412.3474

[11] Valentijn T., Margutti J., van den Homberg M., Laaksonen J. Multi-hazard and spatial transferability of a CNN for automated building damage assessment. Remote Sensing, 12 (17):2839, 2020. URL https://www.mdpi.com/2072-4292/12/17/2839.

[12] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., & Xiao, B. Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 2020. URL https://ieeexplore.ieee.org/abstract/document/9052469.

[13] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. A Comprehensive Survey on Transfer Learning. Proceedings of IEEE, 2020, pp. 43-76.

A Appendix

Table 7: Complete table of F1 scores across different disasters (most damaged training set)

Model	Total F1	Damage F1	Loc F1	No damage F1	Minor Damage F1	Major Damage F1	Destroyed F1
Baseline (tornado)	0.4086	0.2646	0.7445	0.3477	0.01379	**0.1612**	0.5357
SWA (tornado)	0.4353	0.3073	0.7340	0.3566	0.08186	0.1277	0.6629
Multi adaBN (tornado)	0.4413	0.3211	0.7218	0.4179	0.1036	0	0.7630
Multi adaBN + SWA (tornado)	**0.4610**	**0.3368**	**0.7504**	**0.4515**	**0.1281**	0	**0.7683**
Baseline (water)	0**.3288**	**0.2012**	0.6266	**0.5452**	**0.1297**	**0.06262**	0.06726
SWA (water)	0.3109	0.1876	0.5987	**0.5452**	0.09390	0.03066	0.0805
Multi adaBN (water)	0.3190	0.1799	0.6437	0.5222	0.03833	0	0.1591
Multi adaBN + SWA (water)	0.3274	0.1842	**0.6615**	0.5143	0.05441	0	**0.1682**
Baseline (fire)	0.4461	**0.3719**	0.6192	0.7043	0.1398	0.01572	**0.6276**
SWA (fire)	**0.4589**	0.3638	0.5890	0.6807	**0.1491**	**0.01591**	0.6094
Multi adaBN (fire)	0.4049	0.3167	0.6106	**0.7050**	0.03158	0	0.5300
Multi adaBN + SWA (fire)	0.4136	0.3253	**0.6196**	0.7036	0.0538	0	0.5458

Table 8: F1 Scores for models trained with the most damaged training set

Model	Total F1	Damage F1	Loc F1	No damage F1	Minor Damage F1	Major Damage F1	Destroyed F1
Baseline (ood)	xV2: 0.2237 Avg F1: 0.4461	xV2: 0.05423 Avg F1: 0.3719	0.6192	0.7043	0.1398	0.01572	**0.6276**
SWA (ood)	xV2: 0.2152 Avg F1: 0.4589	xV2: 0.05505 Avg F1: 0.3638	0.589	0.6807	**0.1491**	**0.01591**	0.6094
Multi adaBN (ood)	xV2: 0.1832 Avg F1: 0.4049	xV2: 4.0*10-6 Avg F1: 0.3167	0.6106	0.705	0.03158	0	0.53
Multi adaBN + SWA (ood)	xV2: 0.1859 Avg F1: 0.4136	xV2: 4.0*10-6 Avg F1: 0.3253	**0.6196**	**0.7036**	0.0538	0	0.5458
Baseline (iid)	xV2: 0.2584 Avg F1: 0.5384	xV2: 0.02656 Avg F1: 0.4266	0.7993	**0.7938**	0.00979	**0.02173**	**0.8812**
SWA (iid)	xV2: 0.2531 Avg F1: 0.5292	xV2: 0.02112 Avg F1: 0.4157	**0.7941**	0.7611	0.01115	0.01036	0.88
Multi adaBN (iid)	xV2: 0.2342 Avg F1: 0.5534	xV2: 4.0 *10-6 Avg F1: 0.4560	0.7807	0.756	**0.191**	0	0.877
Multi adaBN + SWA (iid)	xV2: 0.2350 Avg F1: 0.5322	xV2: 4.0*10-6 Avg F1: 0.4245	0.7834	0.7436	0.08386	0	0.8707

Table 9: F1 scores for models trained with the balanced dataset.

Model	Total F1	Damage F1	Loc F1	No damage F1	Minor Damage F1	Major Damage F1	Destroyed F1
Multi adaBN ONLY (ood)	xV2: 0.1499 Avg F1: 0.3648	xV2: 4.0*10-6 Avg F1: 0.3070	**0.4995**	0.5783	0.001402	0	0.6483
Multi adaBN + SWA (ood)	xV2: 0.1365 Avg F1: 0.4131	xV2: 4.0*10-6 Avg F1: 0.3950	0.4552	**0.7705**	0.09085	0	**0.7188**
No SWA or multi adaBN (ood)	xV2: 0.1072 Avg F1: 0.3565	xV2: 4.0*10-6 Avg F1: 0.3562	0.3573	0.6043	**0.1581**	0	0.6624
SWA ONLY (ood)	xV2: 0.1283 Avg F1: 0.3804	xV2: 4.0*10-6 Avg F1: 0.3601	0.4276	0.6537	0.1316	0	0.655
multi adaBN ONLY (iid)	xV2: 0.1959 Avg F1: 0.5000	xV2: 4.0*10-6 Avg F1: 0.4344	0.6531	**0.6633**	**0.176**	0	0.8981
Multi adaBN + SWA (iid)	xV2: 0.1925 Avg F1: 0.4573	xV2: 4.0*10-6 Avg F1: 0.3783	0.6415	0.6089	0.01137	0	0.8928
No SWA or multi adaBN (iid)	xV2: 0.2114 Avg F1: 0.4819	xV2: 2.0*10-6 Avg F1: 0.3864	**0.7046**	0.6521	0	0	0.8936
SWA ONLY (iid)	xV2: 0.2071 Avg F1: 0.4784	xV2: 2.0*10-6 Avg F1: 0.3877	0.6901	0.6504	0	0	**0.9003**

Table 10: Confusion matrix localization data for two sets of training data

	True Negative	False Negative	True Positive	False Positive	Recall	Precision
No multiadaBN, no SWA, most damaged	0.95	0.013	0.022	0.014	62.85%	**61.11%**
No multiadaBN, no SWA, balanced	0.89	0.072	0.023	0.012	24.21%	65.71%
SWA, most damaged	0.95	0.012	0.02	0.016	62.50%	55.56%
SWA, balanced	0.91	0.054	0.024	0.011	30.77%	68.57%
multiadaBN, most damaged	0.95	0.0099	0.02	0.016	66.89%	55.56%
multiadaBN, balanced	0.9	0.066	0.03	0.0056	31.25%	**84.27%**
multiadaBN + SWA, most damage	0.96	0.0084	0.02	0.016	**70.42%**	55.56%
multiadaBN + SWA, balanced	0.92	0.048	0.028	0.0079	**36.84%**	77.99%