Towards Post-disaster Damage Assessment using Deep Transfer Learning and GAN-based Data Augmentation

Sourasekhar Banerjee
sourasb@cs.umu.se
Umeå University
Sweden

Yashwant Singh Patel
yashwant.patel@thapar.edu
Thapar Institute of Engineering & Technology
India

Pushkar Kumar
pushkar.btech.ec17@iiitranchi.ac.in
Indian Institute of Information Technology Ranchi
India

Monowar Bhuyan
monowar@cs.umu.se
Umeå University
Sweden

ABSTRACT

Cyber-physical disaster systems (CPDS) are a new cyber-physical application that collects physical realm measurements from IoT devices and sends them to the edge for damage severity analysis of impacted sites in the aftermath of a large-scale disaster. However, the lack of effective machine learning paradigms and the data and device heterogeneity of edge devices pose significant challenges in disaster damage assessment (DDA). To address these issues, we propose a generative adversarial network (GAN) and a lightweight, deep transfer learning-enabled, fine-tuned machine learning pipeline to reduce overall sensing error and improve the model's performance. In this paper, we applied several combinations of GANs (i.e., DCGAN, DiscoGAN, ProGAN, and Cycle-GAN) to generate fake images of the disaster. After that, three pre-trained models: VGG19, ResNet18, and DenseNet121, with deep transfer learning, are applied to classify the images of the disaster. We observed that the ResNet18 is the most pertinent model to achieve a test accuracy of 88.81%. With the experiments on real-world DDA applications, we have visualized the damage severity of disaster-impacted sites using different types of Class Activation Mapping (CAM) techniques, namely Grad-CAM++, Guided Grad-Cam, & Score-CAM. Finally, using k-means clustering, we have obtained the scatter plots to measure the damage severity into no damage, mild damage, and severe damage categories in the generated heat maps.

CCS CONCEPTS

• **Computer systems organization** → *Embedded and cyber-physical systems*; • **Computing methodologies** → *Deep learning*.

KEYWORDS

Cyber-Physical Systems, Generative Adversarial Networks, Deep Learning, Damage Assessment, Class Activation Mapping, Clustering.

ICDCN 2023, January 4–7, 2023, Kharagpur, India

ACM ISBN 978-1-4503-9796-4/23/01.
https://doi.org/10.1145/3571306.3571438

ACM Reference Format:
Sourasekhar Banerjee, Yashwant Singh Patel, Pushkar Kumar, and Monowar Bhuyan. 2023. Towards Post-disaster Damage Assessment using Deep Transfer Learning and GAN-based Data Augmentation. In *24th International Conference on Distributed Computing and Networking (ICDCN 2023), January 4–7, 2023, Kharagpur, India.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3571306.3571438

1 INTRODUCTION

Cyber-Physical Systems (CPS) tightly-integrate networking and computational algorithms into real-world physical objects to offer variety of multidisciplinary applications in the fields of transportation, military, smart environment, healthcare, agriculture, manufacturing industry, smart grid, distributed robotics, process control systems, disaster, and emergency response management [4], [8]. Social sensing based emergency frameworks for disaster systems are designed for generating post-disaster damage assessment reports after major events such as cyclone, hurricane, earthquake, wildfire [12], etc. Afterwards, the emergency rescue teams analyzed the reports of the impacted sites for real-time geographic information systems (GIS) and situational awareness. A typical CPDS with the components of sensing, computation, communication, and control is given in Figure 1.

In current years, computer vision based autonomous damage assessment has become a research hotspot with significant development in deep learning. The most popular types of deep learning algorithms have great advantages and generate more sophisticated results for different computer vision and machine learning (ML) tasks. However, deep learning algorithms suffer from weak generalization when trained on limited data. Convolutional neural networks (CNNs) [9] have significantly improved the performance for several applications [5]. For example, Patel et al. [8] integrated CNN with MobileNetV2 and gradient weighted class activation mapping (Grad-CAM++) to locate and quantify the damages. Banerjee et al. [1] designed a CNN-based deep learning model for classification of multiple diseases from chest-X-ray in a federated setting. The authors show that in federated environment, the ResNet18 model achieves up to 98.3% accuracy during pneumonia detection. The main advantage of CNNs is that they can automatically extract relevant features while mitigating the risk of over-fitting and are also computationally efficient [6]. However, it requires large training data and also don't well encode the position and orientation of objects. Otherwise, it will suffer from data insufficiency problem that

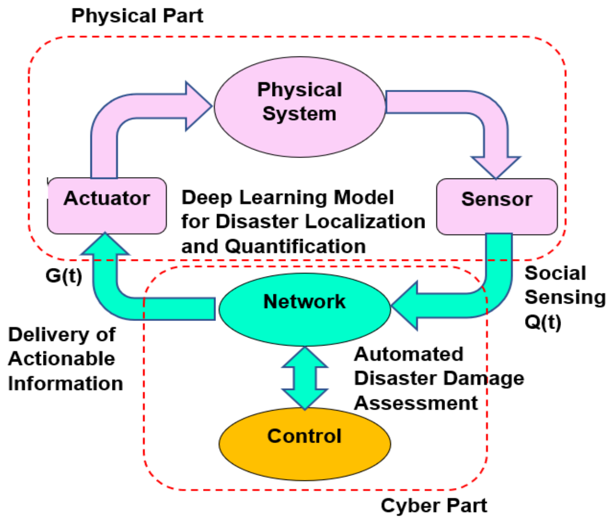

Figure 1: Smart Cyber-Physical Disaster Systems

causes model overfitting. Generative adversarial networks (GANs) are much faster than CNN and robust for learning deep representation without extensively annotated training data [3]. GANs have produced tremendous performance for diverse applications such as image synthesis, super-resolution and classification, and pattern transfer [7]. These benefits of deep learning models are due to data explosion and fast computation. In fact, edge computing systems are started being combined with deep learning for realization of edge intelligence which can offer high resource utilization and throughput [6]. For visual interpretation of the results, CAM based methods such as Score-CAM [11], Grad-CAM [10], and Grad-CAM++ [2] are widely applied for different domains, such as disaster management [8] and healthcare [1]. In this paper, we focus on deep transfer learning based model training and GAN based data augmentation to elevate the problem of model over-fitting, and for assessing the damage severity from disaster images. The contributions of that works are as follows,

1. We developed a GAN and deep transfer learning based machine learning pipeline for disaster damage assessment.
2. The training of the CNN on disaster datasets using transfer learning to classify 6 different types of disasters (e.g., earthquake, wildfire, tsunami, avalanche, drought, thunderstorm)
3. Visualization with CAM based techniques (i.e., Grad-CAM++, Guided Grad-Cam, and Score-CAM) to demonstrate the efficacy of the proposed method for assessing the damage severity (DES) from disaster images.
4. Clustering on DES scores to find the severity level of the disaster image into none, mild, and severe classes.

2 PROBLEM FORMULATION

The disaster identification problem consists: (1) GAN based data augmentation, (2) learning models, (3) image localization, and damage assessment. GAN-based techniques are applied in the data augmentation while deep transfer learning is used to train learning models. Class activation mapping (CAM) based methods are applied for image localization and damage severity assessment (DES).

2.1 Data augmentation

The GAN network incorporates generator $\mathcal{G}_\theta(z)$ and a discriminator $\mathcal{D}_\phi(x)$. Where x denotes samples (either from real images or generator), and z denotes noise vector. The generator and discriminator both play a two-player min-max game. The generator tries to fool the discriminator by generating samples that are indistinguishable from p_{img}. The Objective of GAN can be written as,

$$\mathcal{V}(\mathcal{G}_\theta, \mathcal{D}_\phi) = \mathbb{E}_{x\sim P_{img}}[\log \mathcal{D}_\phi(x)] + \mathbb{E}_{z\sim P(z)}[\log(1-\mathcal{D}_\phi(\mathcal{G}_\theta(z)))] \tag{1}$$

$$\min_\theta \max_\phi \mathcal{V}(\mathcal{G}_\theta, \mathcal{D}_\phi) \tag{2}$$

2.2 Learning problem

Let the pretrained model is $\mathcal{M}(x, w)$. The disaster image dataset contains real and generated images of different natural disasters. Suppose, $\mathcal{X}$ contains input images $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, and $\mathcal{Y} \in \{0, k\}^{1\times n}$ contains corresponding labels. The training data is divided into mini-batches of size, $\mathcal{B}$. The empirical loss function can be defined as:

$$\mathcal{L}(w, X) = \frac{\sum_{x\in X, y\in Y} l(\mathcal{M}(x, w), y)}{B} \tag{3}$$

where y is the true label for input x. w is the model update and $l(.)$ is the multi-class loss function on each sample of B. The objective is to generate a optimal model (w^*).

$$w^* = \arg\min_{w_i\in w} \mathcal{L}(w_i, X) \tag{4}$$

2.3 Localization problem

The localization problem is to detect damage severity from the disaster image efficiently and to identify the affected regions. Localization problem consists of generating heatmaps and computing the Damage assessment scores. First we compute the gradient of feature maps H of a convolutional layer that is used by CAM to localize disaster images. The aggregation of all gradients of output Y^s regarding $H^k_{(i,j)}$, $\forall i, j$ and Z represents the weights w^c_k as the number of pixels in the activation map.

$$w^c_k = \frac{1}{Z}\sum_i\sum_j \frac{\partial Y^s}{\partial H^k_{(i,j)}} \tag{5}$$

Heatmaps are a weighted composite of the created feature maps, preceded by a ReLU function.

$$q_{i,j} = ReLU\Big(\sum_k w^c_k H^k\Big) \tag{6}$$

However, when an picture has many occurrences of the same class, the Grad-CAM fails to execute localization appropriately. Grad-CAM++ solves this problem by taking a weighted average of the

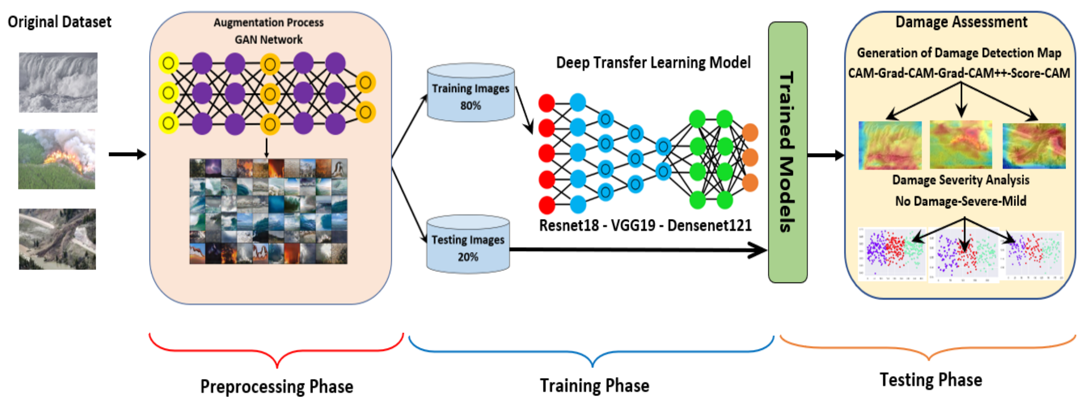

Figure 2: Proposed Machine Learning Pipeline

pixel-wise gradients.

$$w_k^c = \sum_i \sum_j \Phi_{i,j}^{kc} \cdot relu\Big(\frac{\partial Y^c}{\partial H_{(i,j)}^k}\Big) \quad (7)$$

where, $\Phi_{i,j}^{kc}$ is the weighted co-efficient. From the heatmap of the image, an assessment score is calculated based on the the Equation (8). We can imply the level of severity (DES) of the disaster (e.g., mild, severe), according to this score.

$$DES = \frac{1}{Z} \sum_i \sum_j q_{i,j} \quad (8)$$

3 PROPOSED APPROACH

Figure 2 depicts the proposed machine learning pipeline. It is divided into three phases, such as, preprocessing, learning, and, model containerization and assessment of the damage.

Algorithm 1: Preprocessing (Data augmentation)

Input: Disaster image dataset D_{real}, and $D_{real} : \mathcal{X} \longrightarrow \mathbb{R}$
Output: Augmented disaster image dataset D_{aug}
Initialize: number of training epochs ($\mathcal{T}$), number of steps applied to discriminator (κ), $\kappa = 1$
for $\tau = 1 \ldots \mathcal{T}$ **do**
  for κ *steps* **do**
    Sample mini-batch of noise samples $(\{z^i | \forall_i^m z^i \in \mathcal{Z}, z^i \sim \mathcal{P}_{z^i}(z^i)\})$ of size m from latent space ($\mathcal{Z}$).
    Sample mini-batch of size m $(\{x^i | \forall_i^m x^i \in \mathcal{X}\})$ from the data generating distribution $p_{img}(x)$
    Update the generator ($\mathcal{G}_\theta$) using the gradient descent step on θ : $\nabla_\phi V(G_\theta, D_\phi) = \frac{1}{m} \nabla_\theta \sum_{i=1}^m \log(1 - \mathcal{D}_\phi(\mathcal{G}_\theta(z^i)))$
    Update the discriminator ($\mathcal{D}_\phi$) using gradient assent step on discriminator parameter ϕ:
    $\nabla_\theta V(\mathcal{G}_\theta, \mathcal{D}_\phi) = \frac{1}{m} \nabla_\phi \sum_{i=1}^m [\log D_\phi(X^i) + \log(1 - \mathcal{D}_\phi(\mathcal{G}_\theta(z^i)))]$

Algorithm 2: Learning

Input: Disaster image dataset $\mathcal{D}$, $X_i \in \mathcal{D}$, $\mathcal{B}$ = mini batch Size,$\mathcal{E}$ =Total Epochs,
Output: Optimal model parameters w^* , Optimal loss function $\mathcal{L}(w^*)$
Initialize : w^0=w^1, λ, β, $\beta 1$, $\beta 2$
for $e = 1; e \le \mathcal{E}; e+ = 1$ **do**
  for $b = 1; b \le \frac{|D_{train}|}{\mathcal{B}}; b+ = 1$ **do**
    Compute model update using ADAM optimizer:
    $m^b = \beta_1 m^{b-1} + (1 - b_1)\nabla^b$
    $v^b = \beta_2 m^{b-1} + (1 - b_2)[\nabla^2]^b$
    $\bar{m}^b = \frac{m^b}{(1-b_1{}^b)}$
    $\bar{v}^b = \frac{v^b}{(1-b_2{}^b)}$
    and then calculate
    $w^b = w^{b-1} - \lambda \frac{\bar{m}^b}{\sqrt{\bar{v}^b}+\epsilon}$
    Compute loss using Equation
    $\mathcal{L}(w_e) = \frac{\sum_{x \in X_i, y \in Y_i} l(\mathcal{M}(x, w^b), y)}{B}$
  end
end
Select the model using
$\mathcal{L}(w^*) = \min_{e=1 \ldots E}(\mathcal{L}(w_e))$
$w^* = \arg_{e=1 \ldots E} \min(\mathcal{L}(w_e))$
return $w^*, \mathcal{L}(w*)$

Algorithm 1 is the GAN based augmentation for image generation. We supplied six different disaster image-set (see Table 1) to the GAN generator. Then we initialize the total number of training epochs ($\mathcal{T}$) and number of steps (κ) applied to discriminator. The generator network $\mathcal{G}$, generates candidates from the latent space and the discriminator network ($\mathcal{D}$) evaluates them. The outcome of the GAN is a new disaster image dataset D_{aug}. Now, the new dataset contains $D \longleftarrow D_{real} + D_{aug}$. In Algorithm 2, we divided the dataset (D) into train (70%), validation (20%), and test(10%) sets.

Algorithm 3: Assessment of damage and model containerization

Input: Disaster Dataset D with respective output label $\mathcal{Y}$, trained model $\mathcal{M}(\mathcal{D})$.
Output: Damage severity assessment (DES) of disaster image.
$DRM(\mathcal{M}(D)) \longleftarrow$ Features extraction of damage recognition map (DRM)
$DES(DRM(\mathcal{M}(D))) \longleftarrow$Damage severity assessment from DRM
$Cluster(DES) \longleftarrow$ Create three clusters based on the DES score of training images.
Return DES

Algorithm 4: Model containerization

Input: Real-time Disaster Dataset D_I, trained model $\mathcal{M}(D)$, Registered edge devices set $E = \{E_1, E_2, \ldots, E_n\}$
Output: Containerized model $C(\mathcal{M}(D))$
$C(\mathcal{M}(D)) \longleftarrow$ Containerize the trained model $\mathcal{M}(D)$
Deploy $C(\mathcal{M}(D))$ at the registered edge devices E.
Run the Containerized model $C(\mathcal{M}(D))$ on real-time captured disaster images $\mathcal{D}_I = \{\mathcal{D}_{I_1}, \mathcal{D}_{I_2}, \ldots \mathcal{D}_{I_\delta}\}$ at edge devices.

We downloaded pre-trained deep learning models (e.g., ResNet18, VGG19, DenseNet121) and applied transfer learning using D. We initialize the weights (w^0) similar to the pre-trained model, and hyper-parameters $\lambda, \beta, \beta_1, \beta_2$. The model is training for a maximum of $\mathcal{E}$ epochs and selecting the best performing model $\mathcal{M}(D)$ from it. In Algorithm 3, we performed the damage assessment using CAM and clustering method. A damage detection map is extracted from there to interpret the damaged area. Heatmaps are used to illustrate the CAM localized categories, which have soft bounds. We applied clustering to identify the severity of the identified images. We divided the identified images into three clusters, and each cluster specifies the range of severity level of the disaster image. The left cluster (minimum centroid) signifies none, and the rightmost cluster (highest centroid) indicates severe damages. The middle cluster represents mild damage assessment. To provide lightweight and leverage computation at proximity of the data sources, we applied model containerization technique. In Algorithm 4, we containerize the trained model $\mathcal{M}(D)$. We used docker mechanism to containerize a model. The containerized model can be deployed on a set of Edge devices $E = \{E_1, \ldots, E_n\}$ to detect disaster images in real-time (D_I) and produce the severity level of them.

4 EXPERIMENTAL RESULTS

We carried out the following tasks for experiments:

(1) **Data collection :** We collected post disaster images of Earthquake, Wildfire, Tsunami, Avalanche, Drought, and Thunderstorm from social media and pubic available datasets. The details of dataset is in table 1.

Table 1: Dataset details

Dataset	No. of Images
Earthquake	192
Wildfire	410
Tsunami	234
Avalanche	278
Drought	240
Thunderstorm	235

Table 2: Hyperparameter settings

Architecture Details	
Parameter	**Value**
Default hidden layers	16, 19, 150, 103, 48
Training step/epochs	13
Dropout rate	0.4
Batch size	128
Loss function	categorical-crossentropy
Optimizer	Adam
Activation function	Relu
Input layer neurons	150528
Final layer neurons	7
Trainable Parameters	821511, 821511, 1607943, 1345799, 2394375
Non-Trainable Parameters	14714688, 20024384, 2257984, 7037504, 21802784

(2) **Data generation :** The collected data are insufficient for training, therefore we applied GAN to generate the data from the disaster images.

(3) **Transfer learning :** On the generated and collected data, we trained deep learning models using transfer learning. The hyperparameter settings of deep learning models are shown in Table 2.

(4) **Damage detection map :** Extract damage detection map features from images using grad-cam++, guided grad-cam, and score cam.

(5) **Grouping based on severity of damages :** We applied k-means clustering techniques on each dataset based on the damage detection score to group the images as none, mild, and severe.

4.1 Results and Discussion

4.1.1 GAN based data augmentation. We used several GAN based methods, such as Cycle-GAN, Pro-GAN, Disco-GAN, and DCGAN. We discarded generated images with the low resolution. A comparison of the outcome of the different GAN is given in Figure 3. Images generated by Cycle-GAN and ProGAN are much better than DCGAN and DiscoGAN, therefore, images generated form Cycle-GAN and ProGAN are added to the original dataset for training.

4.1.2 Transfer learning. Here we compared 3 pre-trained models ResNet18, VGG19, DenseNet121 on disaster image data sets. We trained these three models for 5 epochs and found that ResNet18 outperforms other methods (see Figure 4). The ResNet18 model produces the loss 0.12 which in comparison to other models is minimum. In Figure 5, we compared the validation accuracy between ResNet18, VGG19, DenseNet121. ResNet18 converges earlier than

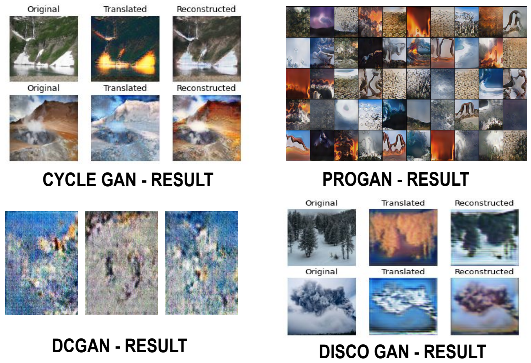

Figure 3: Comparision between Cycle GAN, ProGAN, DCGAN, and DiscoGAN on disaster images

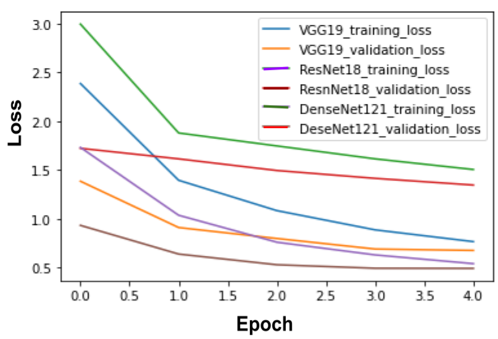

Figure 4: Convergence comparison of VGG19, ResNet18 and DenseNet121 models on disaster images

other methods and produce accuracy of 88.81%. Also, ResNet 18 is a lightweight model that reduces the network depth while widening residual networks.

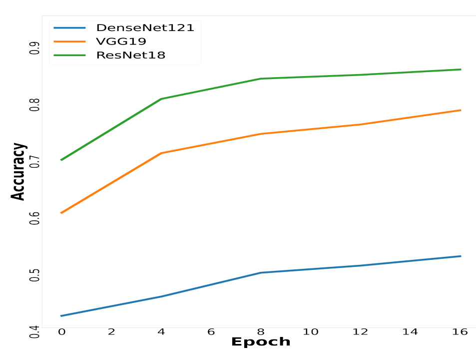

Figure 5: Comparison of validation accuracy of VGG19, ResNet18, and DenseNet121 on disaster image set.

Figure 6 shows the performance of VGG19, DenseNet121, and ResNet18 models across test accuracy, precision, recall, and F1-score. ResNet18 performs best amongst others.

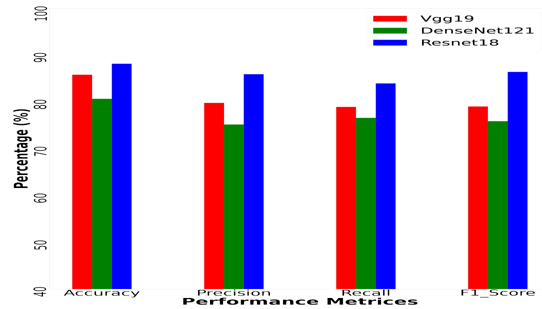

Figure 6: Performance of VGG19, DenseNet121, and ResNet18 models trained on disaster images

4.1.3 CAM-based analysis. In Figure 7, we compared three CAM based techniques (grad-cam++, guided grad cam, and score cam) on disaster datasets. We found that Score-cam outperforms other methods to identify the disaster affected regions.

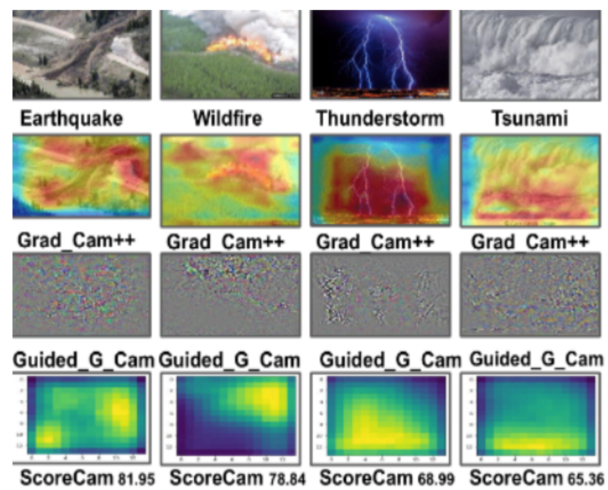

Figure 7: Comparison of different CAM based approach on disaster images

4.1.4 Damage severity analysis. We calculated the level of severity (DES) of each disaster images. And based on the DES scores, we can determine the severity of the disaster. High DES score means Sever damage, whereas, low DES means no-damage. DES score $\in (0, 1)$ Figure 8, shows clustering of images based on severity of the disaster. We applied k-means clustering on DES scores and got 3 clusters such as none, mild, and severe(see in Figure 8). A cluster with pink dots signifies the severely damaged images. red dots implies the disaster is mild, and blue dots represent negligible severity, i.e., no damage. Figure 9, shows how many images fall into these three categories.

5 CONCLUSION

We proposed a fine-tuned deep transfer learning approach for CPDS that takes advantage of GAN to get rid of the data insufficiency problem. We have validated the performance by training three

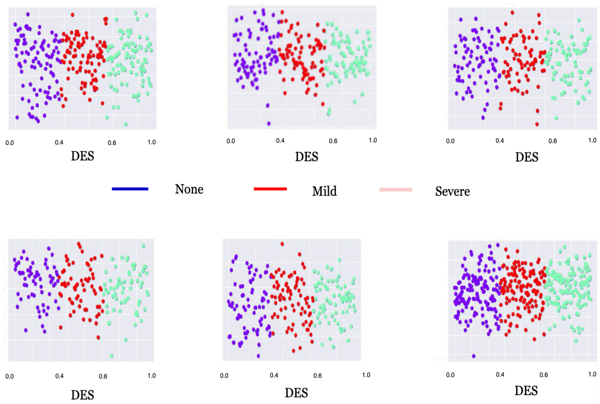

Figure 8: Damage severity analysis, Earthquake (Top left), Wildfire (Top middle), Tsunami (Top right), Avalanche (Bottom left), Drought (Bottom middle), and Thunder storm (Bottom right)

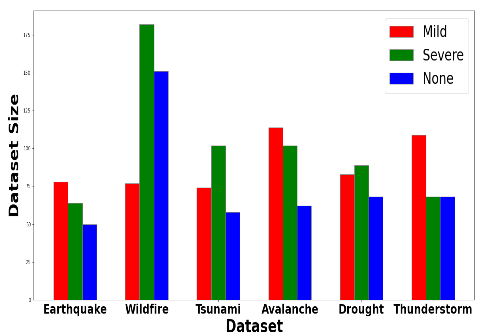

Figure 9: Dataset details

models, i.e., VGG19, DenseNet121, and ResNet18. We observed that ResNet18 is the most pertinent deep transfer model for the real-world disaster damage detection applications. To visualize the severity of damaged sites, we applied CAM enabled techniques, including Grad-CAM++, Guided Grad-Cam, and Score-CAM on trained ResNet18 model and extracted the damage detection maps. Finally, the clustering technique is applied to analyze the level of damage severity into 3 classes named as, severe,mild and no damage. In future, we would like to extend it towards distributed learning.

ACKNOWLEDGMENTS

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundations. This work was also partially supported by Thapar Institute of Engineering and Technology, Patiala, India under SEED research grant.

REFERENCES

[1] Sourasekhar Banerjee, Rajiv Misra, Mukesh Prasad, Erik Elmroth, and Monowar H Bhuyan. 2020. Multi-diseases Classification from Chest-X-ray: A Federated Deep Learning Approach. In *Australasian Joint Conference on Artificial Intelligence*. Springer, 3–15.

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 839–847.

[3] Yizhou Chen, Xu-Hua Yang, Zihan Wei, Ali Asghar Heidari, Nenggan Zheng, Zhicheng Li, Huiling Chen, Haigen Hu, Qianwei Zhou, and Qiu Guan. 2022. Generative adversarial networks in medical image augmentation: a review. *Computers in Biology and Medicine* (2022), 105382.

[4] Marco Conti, Sajal K. Das, Chatschik Bisdikian, Mohan Kumar, Lionel M. Ni, Andrea Passarella, George Roussos, Gerhard Tröster, Gene Tsudik, and Franco Zambonelli. 2012. Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber–physical convergence. *Pervasive and Mobile Computing* 8, 1 (2012), 2 – 21.

[5] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. 2018. Recent advances in convolutional neural networks. *Pattern recognition* 77 (2018), 354–377.

[6] Yiwen Han, Xiaofei Wang, Victor C. M. Leung, Dusit Tao Niyato, Xueqiang Yan, and Xu Chen. 2020. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* 22 (2020), 869–904.

[7] Nour Eldeen M. Khalifa, Mohamed Hamed N. Taha, Aboul Ella Hassanien, and Sally Elghamrawy. 2020. Detection of Coronavirus (COVID-19) Associated Pneumonia based on Generative Adversarial Networks and a Fine-Tuned Deep Transfer Learning Model using Chest X-ray Dataset. arXiv:eess.IV/2004.01184

[8] Yashwant Singh Patel, Sourasekhar Banerjee, Rajiv Misra, and Sajal K. Das. 2020. Low-Latency Energy-Efficient Cyber-Physical Disaster System Using Edge Deep Learning. In *Proceedings of the 21st International Conference on Distributed Computing and Networking (ICDCN 2020)*. Association for Computing Machinery, New York, NY, USA, Article 34, 6 pages.

[9] Claudio Filipi Gonçalves dos Santos and João Paulo Papa. 2022. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Computing Surveys (CSUR)* (2022).

[10] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 618–626.

[11] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu. 2020. Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE Computer Society, Los Alamitos, CA, USA, 111–119. https://doi.ieeecomputersociety.org/10.1109/CVPRW50498.2020.00020

[12] Daniel Zhang, Nathan Vance, and Dong Wang. 2019. When Social Sensing Meets Edge Computing: Vision and Challenges. *arXiv e-prints*, Article arXiv:1905.07528 (May 2019), arXiv:1905.07528 pages. arXiv:cs.HC/1905.07528