Ground-level Post-Disaster Image Classification using DenseNet201 for Disaster Damage Assessment

Angelica Denise J. Abadicio
Manuel S. Enverga University Foundation
College of Computing and Multimedia Studies
Lucena City, Quezon, Philippines
aj.abadicio@mseuf.edu.ph

King Eluard E. Camota
Manuel S. Enverga University Foundation
College of Computing and Multimedia Studies
Lucena City, Quezon, Philippines
ke.camota@mseuf.edu.ph

Kyle Anthony F. Niosco
Manuel S. Enverga University Foundation
College of Computing and Multimedia Studies
Lucena City, Quezon, Philippines
kf.niosco@mseuf.edu.ph

Donabell S. Hernandez
Manuel S. Enverga University Foundation
College of Computing and Multimedia Studies
Lucena City, Quezon, Philippines
donna.hernandez@mseuf.edu.ph

Rodrigo C. Belleza, Jr.
Manuel S. Enverga University Foundation
College of Computing and Multimedia Studies
Lucena City, Quezon, Philippines
rodrigo.belleza@mseuf.edu.ph

Roselyn A. Maaño
Manuel S. Enverga University Foundation
College of Computing and Multimedia Studies
Lucena City, Quezon, Philippines
roselyn.maano@mseuf.edu.ph

David Eric S. Oreta
Manuel S. Enverga University Foundation
College of Computing and Multimedia Studies
Lucena City, Quezon, Philippines
deoreta@mseuf.edu.ph

2023 International Conference On Cyber Management And Engineering (CyMaEn) | 978-1-6654-9329-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/CyMaEn57228.2023.10050981

***Abstract* -** Damage assessment is a quick way for emergency management agencies to figure out the effects of a natural disaster or other remarkable events so that resources can be sent immediately to help with response and recovery. Local officials analyze damage to public and private property after an incident. During the assessment, information is gathered to see if expenses and losses caused by the incident are eligible for help. Volume and degree of building damage are crucial for rescue and recovery; thus, locating damaged areas quickly and correctly. Damage assessment after a natural disaster is time-consuming. This research aims to determine and assess the damage caused by a disaster by developing a model to classify the level of damage (partially or totally damaged) in a structure. The model was based on the DenseNet201 architecture, and different image datasets were used to test it. The model correctly predicts classes with an accuracy of 90%. This classification model was enhanced further through processes called feature extraction and fine-tuning. Fine-tuning resulted in a significant improvement, as evidenced by a gain in training and validation accuracy and a decline in training and validation losses, as demonstrated in the model's learning curve.

Index Terms – *Artificial Intelligence, computer vision, convolutional neural network, disaster damage assessment, transfer learning*

I. Introduction

Natural hazards are unpredictable and can cause catastrophic damage and socioeconomic loss [1]. The Philippines ranked 7 (seven) as the country with the most natural disaster in 2020 [2]. Because of the Philippines' location on the Pacific Ring of Fire, on an active tectonic plate boundary, and in the Typhoon Belt, it is vulnerable to a variety of geological and climatic hazard cascades [3] which is why it is regarded as one of the world's most susceptible and disaster-prone countries [4]. Typhoons, floods, landslides, and wind gusts are all common climate change threats in the Philippines [5]. Disasters are uncontrollable, and their impact is also inevitable [6]. Disasters have a significant impact on life and the economy, as well as infrastructure [7]. From 2010-2019, the Philippine Statistics Authority recorded a total loss of approximately 106 billion pesos of infrastructure damage due to disasters [8]. Damage assessment of a structural building after a disaster is necessary; however, a tedious and slow process [9]. It takes time to evaluate an area for each architectural building and look for damage that natural disasters may have caused. Based on the extent of the damage, it is classified into a subjective rating ('partially damaged' and 'totally damaged') [10]. Minor damage to walls, such as small cracks or loose roofs, is considered "partially damaged." Complete destruction of the infrastructure is considered "totally damaged."

Artificial Intelligence (AI) and machine learning are vital technical components of societal transformation and significantly impact research into social responses to risks and disasters. These technologies are vital for effective disaster management, enabling a quicker and more efficient response. After a disaster, recovery efforts, including community development aid programs, are launched to restore the community's livelihood and assess the damage. The goal is to provide first aid and humanitarian assistance while evaluating the initial damage.

This study is focused on developing a model for assessing infrastructure damage after disasters using image classification, a technique for categorizing and labeling groups of pixels or vectors within an image according to predetermined rules [11]. The highlights are the valuable application of image classification despite its challenges and limitations, including the need for large amounts of annotated data and the potential for biases in the training dataset to impact CNN performance. The model used AI and computer vision to evaluate and classify the severity of damages. Image classification is widely used in various fields, including environmental research, military applications, hydrological science, agriculture, and remote sensing [12]. The model classifies post-disaster images from a ground-level perspective and one household per image, and the full extent of the damage must be visible in the image for it to function correctly. Image classification has been used in various studies, such as "Deep convolutional neural network based medical image classification for disease diagnosis" [13], which used CNN to classify medical images with high accuracy.

II. METHODOLOGY

A. *Comparison of classification models*

Five different architectures were compared using the ground-level post dataset to determine the most suitable one for creating the model. Table 1 shows the two classes' precision, recall, and F1-score of all algorithms. The DenseNet201 algorithm had the highest accuracy at 90%. An entirely connected layer receives input from all previous layers and transmits its feature maps to all subsequent layers [14][15]. Other algorithms tested were the VGG-19, MobileNet, and EfficientNetV2S. The VGG-19 is an algorithm with 16 (3x3) convolutional layers stacked on top of each other and 19 pre-trained weight layers on ImageNet [16], and it also has three fully connected layers. MobileNet is a compact, low-latency, low-power model designed to maximize accuracy while considering resource constraints for on-device or embedded applications [17]. EfficientNetV2S is an architecture that uses Fused MB-Conv for the first three stages and MB-Conv for the remaining stages, resulting in higher performance and shorter training time [18]. The accuracy results for these algorithms - VGG-19, MobileNet, and EfficientNetV2S - were 81%. The performance of the Amazon Rekognition Image was also evaluated on the gathered dataset. Amazon Rekognition is a deep learning-based image recognition service that offers computer vision capabilities and extracts information and insights from images [19]. However, its accuracy score was only 46%, and it was not suitable for use as the algorithm for the developed model.

TABLE I
COMPARISON OF ALGORITHMS

Algorithms Used	Precision		Recall		F1 Score		Accuracy
	0	1	0	1	0	1	
VGG-19	0.82	0.80	0.82	0.80	0.82	0.80	0.81
MobileNet	0.89	0.75	0.73	0.90	0.80	0.82	0.81
EfficientNetV2S	0.89	0.75	0.73	0.90	0.80	0.82	0.81
DenseNet201	**1.00**	**0.83**	**0.82**	**1.00**	**0.90**	**0.91**	**0.90**
AWS Rekognition	0.46	0.19	0.50	0.41	0.48	0.26	0.46

In the study "Performance Comparison of Hybrid CNN-XGBoost and CNN-LightGBM. Methods in Pneumonia Detection," the performance of two hybrid machine-learning algorithms for detecting pneumonia in medical images was evaluated. Both methods performed similarly, indicating that either could be a good choice depending on the user's needs and resources. It is essential to consider the benefits and limitations of each method, such as algorithm complexity and processing requirements, as well as the availability of training data and other resources.

B. *Modeling using DenseNet201*

Datasets such as ImageNet and CIFAR-100 have shown that DenseNet201 performs exceptionally well. DenseNet201 uses a condensed network to create models that are easy to train and highly efficient. Its architecture, which treats feature maps as the network's overall state, allows for feature reuse by multiple layers, increasing the diversity of inputs to later layers and improving performance. Even with a slower growth rate, DenseNet201 performs well due to its ability to access all previous-level feature maps from the layer above it [20].

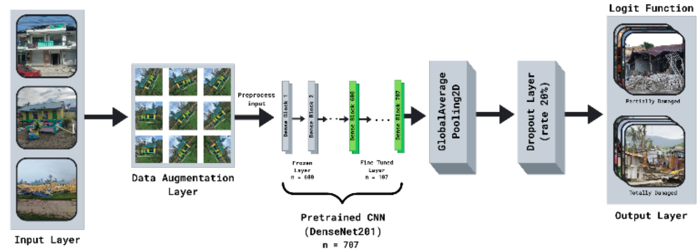

Fig. 1 Classification Model Architecture

The input layer of DenseNet201 is the first layer and only requires the image shape (160, 160, 3) as a parameter as shown in fig.1. The sequential layer following the input layer organizes a stack of layers into a linear structure [21]. Data augmentation, a technique for enhancing a small image dataset, was applied in this layer. Data augmentation involves artificially altering training images in a realistic random manner. In the sequential layer, there are two preprocessing layers called random flip and random rotation that randomly flip images horizontally and rotate them by a factor of two [22] during training. This helps reduce overfitting by exposing the model to different aspects of the training data. The DenseNet201 architecture consisted of various layers and was chosen as the pre-trained CNN for this model, which has 707 layers. The model's performance was improved by fine-tuning the layers by unfreezing them, in this case, after 600 layers. The GlobalAveragePooling2D layer converts the features from their typical 5x5 spatial positions to a single 1280-element vector per image. The dropout layer helps prevent overfitting by randomly setting 20% of the input units to 0 during training. The final layer, the dense output layer, converts each image's learned attributes into a single prediction, with class 0 being predicted as partially damaged and class 1 being predicted as totally damaged. In this case, there is no need for an activation function because the prediction will be taken as a logit or raw prediction value.

C. *Transfer Learning*

Image classification accurately identifies the features present in an image [23], such as those the model learned from partially and totally damaged data. A classifier is created using supervised or inductive machine learning [24], which involves learning from a set of labeled sample characteristics and generalizing them to new cases [25]. Each training set's data must be labeled appropriately, which is common in classification problems [26]. Transfer learning is a machine learning technique that uses the feature representations from a previously trained artificial neural network model to train a new target model on a smaller dataset. It applies the knowledge and information learned from deep learning models to a new task, reducing the need for a large amount of training data and improving the new model's accuracy. Transfer learning is preferred by researchers and data scientists because it allows them to start with a pre-trained model that already understands how to classify objects and has learned the general features of an object [27], leading to better performance compared to training with small data.

D. Data Preprocessing

The data for this study consists of 105 ground-level post-disaster images collected from Disaster Risk Reduction and Management Council offices in Quezon Province and public uploads from various social media platforms. These images show infrastructure damage caused by a disaster. The data was split into train, validation, and test sets, with 80% of the data assigned to the train and validation sets and the remaining 20% assigned to the test set. The data assigned to the train and validation sets were further split, with 60% going to the train set and 40% going to the validation set. The model's training data is stored in the cloud. The pre-processing technique used in this study involves artificially changing the training images through horizontal flipping and rotation to introduce sample variety and reduce overfitting [28]. Data augmentation was also used to enhance the training images.

Data augmentation artificially changes the training images in a realistic, random manner to enhance the training set. This study used data augmentation to increase the model's input data set by horizontally flipping the input data and applying a random rotation of -20% to +20% using the RandomFlip() and RandomRotation() functions. The effect of data augmentation on the data was visualized using the matplotlib tool. The prefetch() method was used to improve the performance of the dataset. The preprocess input function included with the model was used to preprocess the image tensor and rescale the pixel values to the desired input range of [0,1] for the base model DenseNet201.

Fig. 2 Training Dataset

The first nine images of the training set were also visualized using Matplotlib, along with their labels in fig. 2. Data prefetching is a technique that helps improve the performance of a machine-learning model by hiding memory latencies [29]. Buffered prefetching loads data into memory before it is needed, allowing the model to train concurrently with data reading. This results in shorter epoch times compared to the traditional approach of loading data. In this study, 105 images were divided into training, validation, and test datasets, with 50, 34, and 21 images belonging to two classes.

E. Model Evaluation

Model evaluation is an essential aspect of model development. It assists in determining which model best reflects our data and how well it will perform in the future [30]. For the model in this study, the performance was evaluated based on precision, recall, F1 score, and accuracy.

TABLE II
MODEL EVALUATION AND ITS FORMULA

Model Evaluation Criteria	Formula
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1 score	$2 * \frac{Precision * Recall}{Precision + Recall}$
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$

Precision measures the relevance of retrieved instances [31][32], while recall measures the number of correct positive predictions [33]. The F1 score combines precision and recall, and accuracy measures the model's ability to recognize correlations and patterns [34][34]. These evaluation metrics help determine the best model and evaluate its effectiveness in classification or prediction.

III. RESULTS AND DISCUSSION

A. DenseNet201 as the base model

The base model for this study is DenseNet201, pre-trained on the ImageNet dataset. The model's top layer is not included for feature extraction, and predictions are made using the GlobalAveragePooling2D layer. Using a prediction layer, the features are then transformed into a single classification per image. The Keras functional API generates the model, and the Binary Cross entropy function is used for binary classification. The model is compiled, and the function summary displays the model's architecture, with the Dense layer having 1.9 thousand trainable parameters and the DenseNet201 base model having 18.3 million frozen parameters. The base model is trainable in fine-tuning, and the top layers of DenseNet201 are trained with the added layer. The base model has 707 layers, 600 of which are frozen, while the remaining layers are unfrozen. The model is recompiled with a lower learning rate to prevent overfitting, and there are 3.8 million trainable parameters and 14.5 million frozen parameters."

B. Model Evaluation

The DenseNet201 model was examined using the following metrics: precision, recall, f1-score, and accuracy. The DenseNet201 algorithm was trained on the data before testing and evaluating the model. The algorithm initially completed 100 epochs. The program produced results for training loss, training accuracy, validation loss, and validation accuracy after 100 epochs.

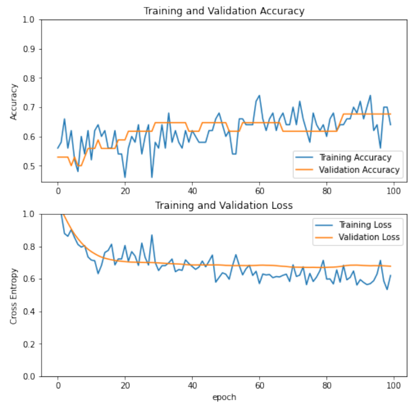

Fig. 3 Learning Curve before Fine Tuning

As shown in fig. 3, the training accuracy of the pre-trained DenseNet201 model started at 0.56 and reached 0.64 after 100 epochs. The validation accuracy increased from 0.56 to 0.68. Both the training and validation losses decreased significantly, with the validation loss reaching 0.68 and the training loss settling at 0.62. Fine-tuning was implemented to improve the algorithm's performance by further training the weights of the top layers of DenseNet201. The procedure began with the 600th of the base model's 707 layers and added 100 epochs.

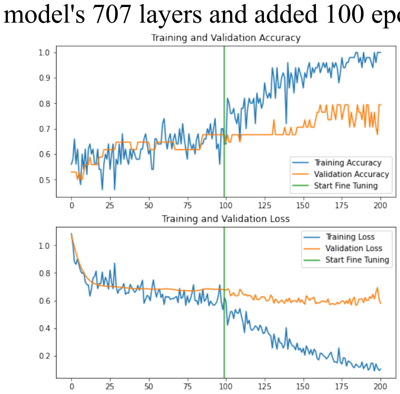

Fig. 4 Learning Curve after Fine Tuning

The effectiveness of fine-tuning is shown by an improvement in training and validation accuracies in Fig. 4, with the training accuracy starting at 0.64 and eventually reaching 1.00. Meanwhile, the validation accuracy increased from 0.65 to 0.79 after fine-tuning. Both the training and validation losses also improved significantly, with the training loss decreasing from 0.59 to 0.10 and the validation loss dropping continuously in an up-and-down pattern.

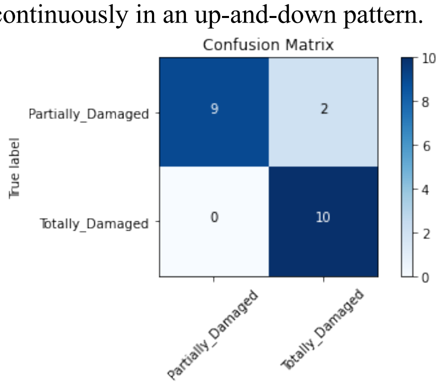

Fig. 5 Confusion Matrix of DenseNet201-based Model

A confusion matrix table summarizes the performance of a classification algorithm [35]. In this study, 21 image data were used to evaluate the model's predictability, as shown in the confusion matrix in Fig. 5. The matrix shows that nine images were correctly predicted as 'partially damaged' and ten were correctly labeled as 'totally damaged,' while two were incorrectly categorized.

	precision	recall	f1-score	support
Partially Damaged (Class 0)	1.00	0.82	0.90	11
Tottally Damaged (Class 1)	0.83	1.00	0.91	10
accuracy			0.90	21
macro avg	0.92	0.91	0.90	21
weighted avg	0.92	0.90	0.90	21

Fig. 6 Classification Report of DenseNet201-based Model

Furthermore, fig. 6 shows the entire classification report of the DenseNet201 model, which includes the evaluation metrics indicated above for each class as well as the overall accuracy of the model. The DenseNet201 model produced precision results of 1.00 and 0.83 for the 'partially damaged' and 'totally damaged' classes, respectively. This means that all predicted values for the 'partially damaged' class were correctly classified, and 83% of the images labeled as 'totally damaged' were actually 'totally damaged.' The model also had high recall values of 0.82 and 1.00 for the 'partially damaged' and 'totally damaged' classes. This means that 82% of the 'partially damaged' images were correctly labeled, and all 'totally damaged' data were accurately identified. The resulting F1 scores of 0.90 and 0.91 for the two classes indicate that the model accurately classifies observations and performs well. The overall accuracy score of 90% shows that only 10% of the data used in the testing phase were incorrectly classified, indicating that the model is performing well in classifying the data. The DenseNet201 algorithm was used to construct the model and was compared to other models from relevant studies, such as DamFormer [36], RescueNet [37], and DCFNet [38], which also focus on building damage assessment.

TABLE III
Comparison to Other Existing Models

Models	Accuracy Score
DamFormer	0.728
RescueNet	0.740
DCFNet	0.800
DenseNet201	**0.905**

C. *Mobile Application*

Python's simple Machine Learning techniques make it easy to analyze datasets and make preliminary predictions. However, for these trained models to be effective in the real world, they must be available on the web or portable devices. TensorFlow Lite is a platform for training Machine Learning models on mobile, IoT, and embedded devices. The trained model can be converted into a TensorFlow Lite flat buffer file ".tflite" for Android and iOS devices. TensorFlow Lite executes all processes on the device, eliminating the need for data transfer to and from a server. The APK was built using Android Studio and can be accessed on Android devices. When the program is launched, as shown in fig. 7, the user is presented with options such as "Take a photo" or "Launch Gallery." Choosing "Take a photo" allows the user to take a ground-level post-disaster image, which the application will classify as 'Partially-damaged' or 'Totally-damaged.' The mobile application developed using the classification model implementation ensures the feasibility of the study.

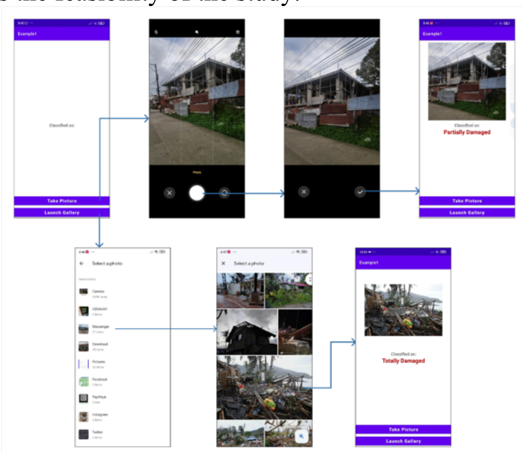

Fig. 7 DenseNet201-based Mobile Application

D. *Relevance to Actual Crisis Management*

Feedback and evaluation were obtained from practitioners in the field of damage assessment; specifically, officers of the DRRM office or council, to ensure that the model fits and performs effectively for operational purposes. The classification model will benefit the agency by improving and speeding up their traditional method of assessing disaster damages in infrastructure. It is recommended that the model be developed into a fully functional application with data inputs and damage cost estimation integrated to standardize data from lower-level barangays and eliminate errors. Based on practitioner feedback, the model's performance in classifying images is satisfactory.

IV. CONCLUSION AND FUTURE WORKS

This study found that developing a model for assessing disaster damage to infrastructure was efficient and beneficial, improving the traditional method and benefiting workers from DRRM offices or councils. The dataset used in the study was small, consisting of 105 images because not all data obtained from DRRMOs in Quezon Province was suitable for the study. However, data pre-processing techniques were applied to generate additional data, resulting in 420 more images. The modified DenseNet201 model was enhanced through feature extraction and fine-tuning and performed optimally, achieving an accuracy of 90%. Fine-tuning also significantly improved the model, as shown by increased training and validation accuracies and decreased training and validation losses. The deployment of the model into a functional mobile application demonstrated the feasibility and realization of the idea. In future work, the mobile app could include a data collection feature for each image, providing household and building information that can be generated into a report. This could be a way of digitalizing the traditional damage assessment approach used by DRRMOs, helping to eliminate data input mistakes and potentially providing a new methodology for assessment. Additionally, there is potential to improve the model by using fuzzy systems, which process information using fuzzy methodologies and are frequently used in circumstances where traditional set theory and binary logic are insufficient [39]. There are several related studies that the study can learn from and be improved upon, such as [39], [40], [41], and [42].

ACKNOWLEDGMENT

Sincerest gratitude and acknowledgment are extended to the College of Computing and Multimedia Studies of Manuel S. Enverga University Foundation for the guidance and support for this study. We would also like to thank Professor Roselyn A. Maaño for her insightful advice and unending help during the writing process. The same could be said for Professor Donabell S. Hernandez, who gave us brilliant suggestions and feedback that helped us enhance our work even further. We would also like to acknowledge the Provincial and Mauban DRRMOs, especially Mr. Harold Reasonda, for their generous support in supplying us with the information and data we needed to complete the study. Their assistance was essential to the success of our project.

REFERENCES

[1] R. J. Andrews and L. M. Quintana, "Unpredictable, unpreventable and impersonal medicine: global disaster response in the 21st century," EPMA Journal, vol. 6, no. 1, p. 2, Dec. 2015, DOI: 10.1186/s13167-014-0024-9.
[2]F. Dell'Acqua and P. Gamba, "Remote sensing and earthquake damage assessment: Experiences, limits, and perspectives," Proceedings of the IEEE, vol. 100, no. 10, pp. 2876–2890, 2012, doi: 10.1109/JPROC.2012.2196404.
[3]L. Goce Minimo, "Spatiotemporal Analysis of the Interaction of Decentralization, Development and Disaster Cascades in Mindanao, Philippines," 2021.
[4]Ma. D. G. Z. Roldan, "Addressing Climate Change and Disaster Risk Reduction through ICT and EU Assistance: The Case of Philippine Local Governance," European Journal of Sustainable Development, vol. 11, no. 2, p. 32, Jun. 2022, DOI: 10.14207/ejsd.2022.v11n2p32.
[5]J. M. S. de Leon and D. B. Magcale-Macandog, "Physical Vulnerability Assessment to Flooding of Residential Houses Along the Coastal Areas in Santa Rosa City, Laguna, Philippines," in Blue-Green Infrastructure Across Asian Countries, Singapore: Springer Singapore, 2022, pp. 357–378. doi: 10.1007/978-981-16-7128-9_16.
[6]Great Learning Team, "What is Artificial Intelligence? How Does AI Work? - Great Learning," Jan. 19, 2022. https://www.mygreatlearning.com/blog/what-is-artificial-intelligence/#HowArtificialIntelligenceworks (accessed May 07, 2022).
[7]J. Hidalgo and A. A. Baez, "Natural Disasters," Crit Care Clin, vol. 35, no. 4, pp. 591–607, Oct. 2019, doi: 10.1016/j.ccc.2019.05.001.
[8]A. Oppermann, "AI vs. Machine Learning vs. Deep Learning: What's the Difference? | Built-In," 2022. https://builtin.com/artificial-intelligence/ai-vs-machine-learning (accessed May 07, 2022).
[9]A. Veronica, S. Gabriel, P. M. Pulhin, and R. D. Lasco, "State of Loss and Damage Assessment System in the Philippines and Proposed L&D Framework."
[10]K. R. Nia and G. Mori, "Building damage assessment using deep learning and ground-level image data," Proceedings - 2017 14th Conference on Computer and Robot Vision, CRV 2017, vol. 2018-Janua, pp. 95–102, 2018, DOI: 10.1109/CRV.2017.54.
[11]I. Redko, A. Habrard, E. Morvant, M. Sebban, and Y. Bennani, "Domain Adaptation Problem," in Advances in Domain Adaption Theory, Elsevier, 2019, pp. 21–36. DOI: 10.1016/B978-1-78548-236-6.50002-7.

[12]M. A. Chandra and S. S. Bedi, “Survey on SVM and their application in image classification,” International Journal of Information Technology, vol. 13, no. 5, pp. 1–11, Oct. 2021, DOI: 10.1007/s41870-017-0080-1.
[13]S. S. Yadav and S. M. Jadhav, “Deep convolutional neural network based medical image classification for disease diagnosis,” J Big Data, vol. 6, no. 1, p. 113, Dec. 2019, DOI: 10.1186/s40537-019-0276-2.
[14]G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.
[15]S. Wang and Y.-D. Zhang, “DenseNet-201-Based Deep Neural Network with Composite Learning Factor and Precomputation for Multiple Sclerosis Classification,” ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 16, pp. 1–19, Jun. 2020, DOI: 10.1145/3341095.
[16]R. Agarwal, “State Classification of Cooking Images Using VGG19 Network.”
[17]S. M. Novikov, A. B. Nurgabylova, D. A. Filimonov, and V. v Poroĭkov, “[The forecasting of hygienic regulations for industrial substances possessing irritant action].,” Gig Sanit, no. 6, pp. 16–20, 1995.
[18]R. Entsuah, A. Derivan, and D. Kikta, “Early onset of antidepressant action of venlafaxine: pattern analysis in intent-to-treat patients.,” Clin Ther, vol. 20, no. 3, pp. 517–26, 1998, DOI: 10.1016/s0149-2918(98)80061-1.
[19]A. Jet and H. J. O, “Supervised Machine Learning Algorithms: Classification and Comparison,” International Journal of Computer Trends and Technology, vol. 48, 2017, DOI: 10.14445/22312803/IJCTT-V48P126.
[20]A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur, “Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning,” J Biomol Struct Dyn, vol. 39, no. 15, pp. 5682–5689, Oct. 2021, DOI: 10.1080/07391102.2020.1788642.
[21]S. S. Nath, G. Mishra, J. Kar, S. Chakraborty, and N. Dey, “A survey of image classification methods and techniques,” in 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Jul. 2014, pp. 554–557. doi: 10.1109/ICCICCT.2014.6993023.
[22]A.Gandhi, “Data Augmentation | How to use Deep Learning when you have Limited Data,” Nanonets AI & Machine Learning Blog, May 2021.
[23]yesilscience, “Transfer Learning Using DenseNet201,” Yesil Science, Mar. 2022, Accessed: Nov. 27, 2022. [Online]. Available: https://www.yesilscience.com/transfer-learning-densenet201/
[24]I. Muhammad and Z. Yan, “SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY,” ICTACT Journal on Soft Computing, vol. 05, no. 03, pp. 946–952, Apr. 2015, doi: 10.21917/ijsc.2015.0133.
[25]“Emerging Artificial Intelligence Applications in Computer Engineering,” Google Books, 2022, Accessed: Dec. 02, 2022. [Online]. Available: https://books.google.com.ph/books?hl=en&lr=&id=vLiTXDHr_sYC&oi=fnd&pg=PA3&dq=supervised+machine+learning&ots=CZqABB_Iir&sig=iET0hj9dut4WtAxUdmcjvq2ESlM&redir_esc=y#v=onepage&q=supervised%20machine%20learning&f=false
[26]C. Zhao, R. Shuai, L. Ma, W. Liu, D. Hu, and M. Wu, “Dermoscopy Image Classification Based on StyleGAN and DenseNet201,” IEEE Access, vol. 9, pp. 8659–8679, 2021, DOI: 10.1109/ACCESS.2021.3049600.
[27]“A Newbie-Friendly Guide to Transfer Learning: Everything You Need to Know.” https://www.v7labs.com/blog/transfer-learning-guide (accessed May 09, 2022).
[28]C. Shorten, T. M. Khoshgoftaar, and B. Furht, “Text Data Augmentation for Deep Learning,” J Big Data, vol. 8, no. 1, p. 101, Dec. 2021, DOI: 10.1186/s40537-021-00492-0.
[29]D. G. M. T. K. and M. A. R. Khanna, “Machine Learning Techniques for Improved Data Prefetching,” ResearchGate, 2015, Accessed: Dec. 08, 2022. [Online]. Available: https://www.researchgate.net/publication/277301694_Machine_Learning_Techniques_for_Improved_Data_Prefetching
[30]D. Wallach, D. Makowski, J. W. Jones, and F. Brun, “Model Evaluation,” Working with Dynamic Crop Models, 2014. https://www.saedsayad.com/model_evaluation.htm (accessed May 07, 2022).
[31]“Practices of Science: Precision vs. Accuracy,” Hawaii.edu, 2022, Accessed: Nov. 27, 2022. [Online]. Available: https://manoa.hawaii.edu/exploringourfluidearth/physical/world-ocean/map-distortion/practices-science-precision-vs-accuracy#:~:text=Precision%20refers%20to%20how%20close,are%20both%20accurate%20and%20precise
[32]A.Shafi, “What are the definitions of Precision and Recall? | Towards Data Science,” Medium, Jan. 2022.
[33]Exsilio Solutions, “, “Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog,” Exsilio Blog, Sep. 2016, Accessed: Nov. 27, 2022. [Online]. Available: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.
[34]Zeya LT, “Essential Things You Need to Know About F1-Score - Towards Data Science,” Medium, Nov. 2021.
[35]A. Kulkarni, D. Chong, and F. A. Batarseh, “Foundations of data imbalance and solutions for a data democracy,” in Data Democracy, Elsevier, 2020, pp. 83–106. DOI: 10.1016/B978-0-12-818366-3.00005-8.
[36]H. Chen, E. Nemni, S. Vallecorsa, X. Li, C. Wu, and L. Bromley, “Dual-Tasks Siamese Transformer Framework for Building Damage Assessment,” Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.10953
[37]R. Gupta and M. Shah, “RescueNet: Joint Building Segmentation and Damage Assessment from Satellite Imagery,” Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.07312
[38]H. Xiao, Y. Peng, H. Tan, and P. Li, “Dynamic Cross Fusion Network for Building-Based Damage Assessment,” in 2021 IEEE International Conference on Multimedia and Expo (ICME), Jul. 2021, pp. 1–6. DOI: 10.1109/ICME51207.2021.9428414.
[39]R. Czabanski, M. Jezewski, and J. Leski, “Introduction to Fuzzy Systems,” 2017, pp. 23–43. doi: 10.1007/978-3-319-59614-3_2.
[40]P. Murugeswari and S. Vijayalakshmi, “New Method of Internal Type-2 Fuzzy-Based CNN for Image Classification,” International Journal of Fuzzy Logic and Intelligent Systems, vol. 20, no. 4, pp. 336–345, 2020, DOI: 10.5391/IJFIS.2020.20.4.336.
[41]H. Sofyan, E. Elfayani, A. Rahmatika, M. Marzuki, and I. Irvanizam, “Applying Fuzzy Decision Tree Method for Hypertension Classification in Adolescent,” 2022, pp. 360–368. DOI: 10.1007/978-3-031-09173-5_44.
[42]H. Habernek, O. Kwasny, L. Schmid, and F. Ortner, “Complications of interlocking nailing for lower leg fractures: a 3-year follow up of 102 cases.,” J Trauma, vol. 33, no. 6, pp. 863–9, Dec. 1992, doi: 10.1097/00005373-199212000-00012.