Received: 16 June 2023 | Revised: 24 July 2023 | Accepted: 9 August 2023 | Published online: 10 August 2023

RESEARCH ARTICLE

Artificial Intelligence and Applications
2024, Vol. 2(2) 121–128
DOI: 10.47852/bonviewAIA32021224

BON VIEW PUBLISHING

Supportive Environment for Better Data Management Stage in the Cycle of ML Process

Lama Alkhaled[1,*] and Taha Khamis[2]

[1]*Department of Computer Science, Electrical and Space Engineering, Lulea University of Technology, Sweden*

[2]*Faculty of Engineering, University of Malaya, Malaysia*

Abstract: The objective of this study is to explore the process of developing artificial intelligence and machine learning (ML) applications to establish an optimal support environment. The primary stages of ML include problem understanding, data management (DM), model building, model deployment, and maintenance. This paper specifically focuses on examining the DM stage of ML development and the challenges it presents, as it is crucial for achieving accurate end models. During this stage, the major obstacle encountered was the scarcity of adequate data for model training, particularly in domains where data confidentiality is a concern. The work aimed to construct and enhance a framework that would assist researchers and developers in addressing the insufficiency of data during the DM stage. The framework incorporates various data augmentation techniques, enabling the generation of new data from the original dataset along with all the required files for detection challenges. This augmentation process improves the overall performance of ML applications by increasing both the quantity and quality of available data, thereby providing the model with the best possible input.

Keywords: machine learning application, data management, data augmentation, insufficient data

1. Introduction

1.1. Background

In today's world, machine learning (ML) applications have become integral to various aspects of our lives (Kim et al., 2018). Whether it is transportation, healthcare, document analysis, or any other industry (Adewumi et al., 2022; Alkhaled et al., 2023; Kanchi et al., 2022; Khan et al., 2022), the proliferation of ML applications has been rapid (Amershi et al., 2019). ML is a fundamental component of artificial intelligence (AI), where data and algorithms are utilized to enable AI systems to mimic human behavior and cognition. The process of teaching AI closely mirrors our own learning process as humans in our daily lives (Brown, 2021). The successful implementation of ML advancements, particularly deep learning, has revolutionized the world in numerous ways (Voon et al., 2022), introducing remarkable applications that are impossible to overlook (Ashmore et al., 2021). The sole difference between traditional programming and ML development is that in traditional programming, we are aiming to solve a specific problem by setting predefined rules or logic as a program while in ML the main aim is to feed the AI with a set of algorithms and data to enable it to learn as a human and answering the question or solving the problem by analyzing the input data (Daisy, 2021). While the development of ML applications continues to grow, several obstacles have come to light (de Souza Nascimento et al., 2019). One of the most challenging stages in the ML development life cycle is the data management (DM) stage. The lack of adequate and relevant datasets significantly impacts the performance of ML models. Inaccurate or insufficient datasets can weaken the accuracy of ML applications, hindering their real-world effectiveness. As such, ensuring a successful start in the early phases of ML, particularly in DM, is crucial to ensuring the accuracy and reliability of the final ML application.

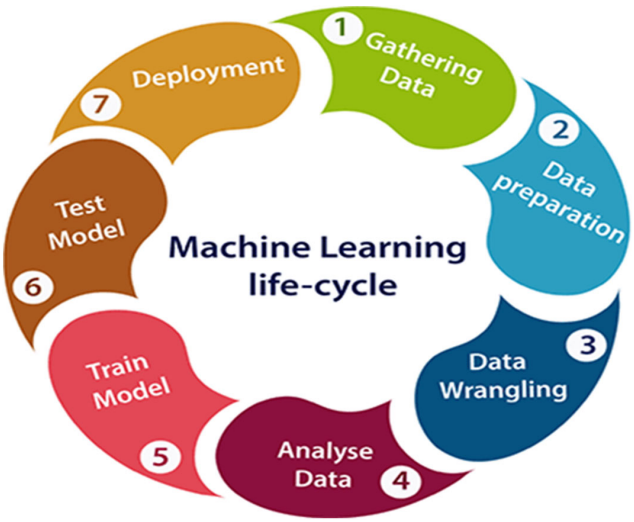
To summarize, a high-level view of the ML application development life cycle can be summarized as follows: first, a problem must be understood; second, data must be collected and preprocessed in a stage called DM; then, a model must be built; and finally, model deployment and maintenance must be performed (Morgunov, 2023) as shown in Figure 1.

1.2. Problem definition

The DM stage of ML development is the most challenging stage, with numerous obstacles arising during this phase (Ashmore et al., 2021). The most significant issue with the model is the paucity of accessible data. The accuracy of ML application outputs might be weakened by using datasets that are either too small or incorrectly formatted and organized. A successful start in the early phases of ML is crucial for the rest of the process. We must guarantee that the early phases of ML, such as DM, are error-free to ensure that the final application is accurate. Data

*Corresponding author: Lama Alkhaled, Department of Computer Science, Electrical and Space Engineering, Lulea University of Technology, Sweden. Email: lama.alkhaled@ltu.se

Figure 1
The machine learning lifecycle



augmentation is one of the strategies that can be used by developers to tackle the problems in the DM stage by feeding the model with produced datasets that increase the accuracy of the output as well as fixing the data availability problems. We can increase the likelihood of a more accurate ML application and decrease the amount of time developers spend attempting to address data problems using conventional ways by building a framework that incorporates the most modern technology.

To address the limitations and challenges of DM in the ML development life cycle, this research aims to streamline the DM stage by creating a framework that enables the implementation of augmentation techniques quickly and effortlessly, eliminating the need for manual coding. The proposed framework empowers users to specify the data path, upload it to the application, and select desired augmentation techniques from a provided list in the application's graphical user interface (GUI). By automating the augmentation process, the framework reduces the time and effort required for data labeling and classification into respective folders, thereby enhancing the overall DM process.

1.3. Scope of the work

With a good framework for DM in the ML development cycle, the amount of time and effort spent on data collecting and cleaning will be decreased which, according to many studies, is a big issue for developers (Roh et al., 2021). There are several methods and strategies available for dealing with the problems that arise during the DM stage (Wong et al., 2016). As a result, developers may have a hard time deciding which approach or method to apply in any given circumstance. Because of this, when the most cutting-edge and efficient technologies are properly used during the DM stage, developers will save both time and effort while boosting their chances of creating more accurate and successful ML apps.

By addressing the research gap in DM for ML applications, this study aims to contribute to the advancement of the ML development process, ensuring more accurate and reliable ML applications. Additionally, this research can benefit developers, researchers, and practitioners in various industries by providing them with a user-friendly framework that automates the data augmentation process, ultimately leading to improved ML model performance.

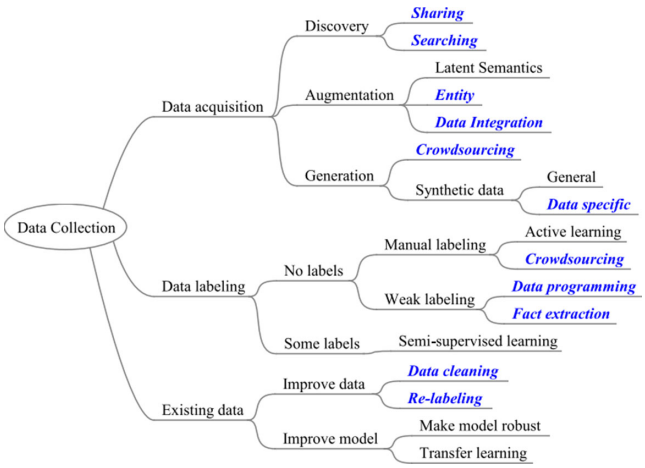
The rest of this paper will discuss the research methodology, the proposed framework, case studies that illustrate its implementation, experimental results, and a conclusion. The conclusion will also discuss the limitations of this study and suggest future research avenues.

2. Literature Review

Data form the essence of most applications in today's world. It is important to note that ensuring the safety and security of our data is a crucial aspect across all types of applications. One effective approach to safeguarding our data involves utilizing cutting-edge tools that minimize the risk associated with network connections (Machap & Khamis, 2022). Research by Douglass (2020) found that gathering data is the initial step in the DM process. The gathering of data samples for use in a ML model by means of observations and measurements is the purpose of data collection (Ros et al., 2016). For example, a student's height or the number of pupils or movies seen may be a numerical data point, a category data point, or even plain text like a student's notes or a student's journal. It is necessary to gather information from many sources to create an AI/ML solution that can be put into practice. Therefore, the data acquired should be relevant to the business issue being addressed. The capacity to gather data allows you to keep a record of earlier occurrences, which may be utilized to find patterns that repeat via the analysis of data acquired. ML algorithms may be used to develop prediction models based on these patterns, which seek trends and forecast future changes. Developing high-quality prediction models necessitates the use of efficient processes for gathering data. The data must be error-free and include the necessary information to guarantee the task's success. According to a recent study (Roh et al., 2021), there are three major ways to gather data depending on the scenario. Sharing and searching for new datasets are the primary goals of data gathering. Discover, augmenting, and producing datasets are some of the methods we might utilize in the data-collecting process. According to the research by Whang and Lee (2020), obtaining the proper datasets for training models is a large difficulty for ML developers since the amount of data accessible throughout the globe is enormous and constantly expanding, making it difficult for ML developers to discover suitable datasets. When the datasets are available, the second procedure will be used: data labeling. Individual data may be labeled using a variety of ways, including manual labeling and weak labeling, in this method. If we can make changes to current data rather than creating new datasets, we may apply the last option mentioned above. It is possible to combine these three approaches, according to Roh et al. (2021) as shown in Figure 2.

We may utilize augmentation techniques if data samples cannot be obtained for any reason or if they are too costly or time-consuming to gather (Ros et al., 2016). Supplementing the acquired data with new samples is accomplished via the use of augmentation techniques (Wong et al., 2016). An additional scenario is when numerous resources' data are not homogenous or need preprocessing to maintain consistency in datasets for validation and verification in different sources. Aside from reducing the complexity of data, preprocessing may also be utilized for training purposes. There may be times when preprocessing is necessary to label data samples for supervised ML tasks (Douglass, 2020). Analyzing data allows for the augmentation and preparation of data to be completed. Following the collection of datasets, several

Figure 2
Data collection methods



actions will be carried out to prepare the data for use in the model-building process. Typically, the first action that will be carried out after the acquisition of data is data labeling. Data labeling is the act of classifying and marking data to better understand its behavior and trends to make better decisions (Borges et al., 2021). When the datasets have been completed, the data labeling process takes over. The semi-supervised strategy is the first approach utilized in data labeling, and it makes use of current labels to predict new labels in the data (Whang & Lee, 2020). Crowdsourcing is an alternate labeling methodology that uses manual labeling and more sophisticated approaches, such as active learning, to finish the labeling process. When implementing crowdsourcing, controlling the quality of processing and post-processing are two of the most difficult issues (Sheng & Zhang, 2019). For certain applications, a small team of in-house information technology professionals might perform better than a big one (Taddeo et al., 2019). Frequently, the person who labels a feature clearly defines the true worth of that characteristic. In other cases, however, this may not be the case: is the cyclist or pedestrian who is riding a bicycle? While ambiguity may be meaningless in certain situations, it is critical in others. Consistencies in labeling are inevitable when the labels are generated by human beings. Finding and correcting discrepancies is an ongoing challenge (Ashmore et al., 2021). Another recent trend is the use of labeling that is too slender. Semi-automatic label production can compensate for poor label quality by mass-producing weak labels. In the absence of numerous labels, simpler regulations are preferred. Rather than searching for and labeling new datasets in some situations, such as when there are no relevant data to employ in a specific application and no relevant data are accessible, we may improve the quality of present datasets. Instead of wasting time and resources on unneeded datasets, the goal is to make the most of what you currently have. As another example, it is better to clean and/or re-labeled old data rather than add new datasets to the model if the additional datasets do not increase the model's overall accuracy since the datasets are of low quality. When it comes to the training data, it is extremely common for there to be some inaccuracies in this data (Whang & Lee, 2020). ML systems, such as "TensorFlow Extended," may aid in the process of validation and cleaning by finding problems in data utilizing data visualization and schema construction approaches, as well as other ways (Baylor et al., 2017). Data cleaning is a technique that may be used to rectify inaccuracies in data, and there is a substantial amount of study on the subject (Ilyas & Chu, 2019).

3. Implementation

3.1. Research methodology

The objective of this research is to streamline the DM stage by creating a framework that enables the implementation of augmentation techniques quickly and effortlessly, eliminating the need for manual coding. The framework allows users to specify the data path, which is then uploaded to the application. Developers or researchers can select the desired augmentation techniques from a list provided on the application's GUI. Upon selecting the augmentation methods, the framework executes them and appropriately labels the resulting images. This reduces the time required for data labeling and classification into respective folders. The flowchart below illustrates the methodology employed in this research and its execution process.

The research consists of four main phases as follows and clarifies in Figure 3:

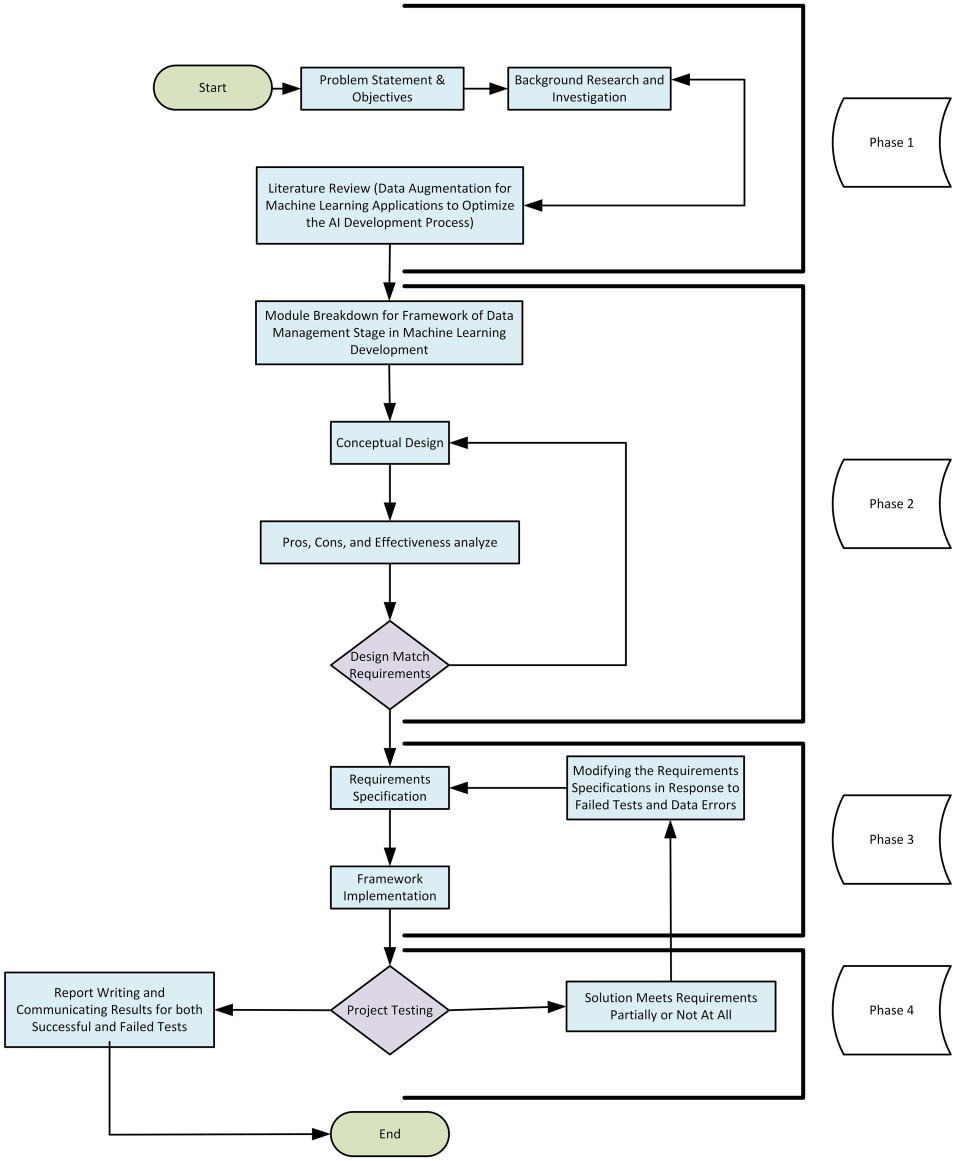
A. Gaining knowledge and understanding of the context. A comprehensive evaluation of the DM stage on the ML development life cycle, the issues encountered, and solutions used in the literature. Analysis of the principal limits and possible issues is then carried out. This phase's primary objective is to get an understanding of current developments in ML.
B. The initialization process of creating a DM framework (conceptual design): Based on the prior stage, a breakdown of modules encompassing requirements, activities, and programming designs will be presented according to the scale of the needed system. Drawings of the flowchart, activity diagram, and necessary designs will be specified, and the programming needed for the functions will be produced.
C. The beginning of the process of creating a DM framework (development): During phase 2, the criteria for the idea will be finalized. After verifying that the ideas adhere to the specifications, the framework will be ready for utilization.
D. Testing and evaluation: The final application will be evaluated based on the results of phases 1 and 2. To improve the framework and address potential difficulties, the data obtained may be utilized. The framework's effectiveness and correctness will be assessed primarily.

3.2. Research method

There are two basic categories of image augmentation techniques: white-box and black-box methods based on deep neural networks. Each has its own set of benefits (Mikolajczyk & Grochowski, 2018). The method of this research is to combine multiple white-box data augmentation techniques. The chosen techniques have proven their success in improving the accuracy of different models and solving the insufficiency of data. The combined techniques are grouped into different categories as follows:

- Geometric transformation (GT): resize and rescale, random rotate and flip, translation, shear, zoom.

Figure 3
Proposed methodology



- Color spaces (CS): random darkness and brightness, CS transformation (greyscale, binary, Hue, Saturation, Lightness (HSV), Cyan, Magenta, Yellow, and Key (CMYK)),
- Kernel filters (KFs): sharpening, blurring,
- Random erasing (RE): random portion,
- Mixing images (MIs): averaging, weighted average,
- Weather simulation (WS): pixel-level transformations to simulate snow, rain, fog, and haze.

To consolidate the desired techniques within a single framework, three widely recognized libraries were utilized: imgaug (Jung et al., 2020), torchvision (Zero & Hug, 2016), and albumentations (Buslaev et al., 2020). These libraries specialize in offering comprehensive image augmentation capabilities specifically designed for computer vision training pipelines. The primary purpose of integrating these libraries is to facilitate regularization and enhance the overall training process.

3.3. Dataset and case study

Two datasets were used to evaluate the state-of-the-art (STOA) object detectors and highlight the advantage of using this framework to propose better data for these models.

3.3.1. Case study 1: Nordic Vehicle Dataset (NVD)

This dataset includes information about vehicles that were captured by unmanned aerial vehicles in diverse environments and varying snow cover conditions within the Nordic region. The data encompass a range of challenging weather conditions such as snowy overcast, low light with patchy snow cover, high brightness, sunlight, fresh snow, and extremely low temperatures below −0° Celsius. NVD consists of 26,313 annotated cars captured from different altitudes (120–250 m) with different snow and lightning conditions (Mokayed et al., 2023).

Table 1
Performance in seconds of different proposed augmentation techniques using different libraries

	Imgaug	Torchvision	Albumentations
GT-rotate	0.019	**0.011**	0.028
KF-Blur	0.13	0.143	**0.005**
CS-greyscale	0.30	0.002	**0.001**
RE	**0.034**	–	–
WS (Snow)	–	–	**0.53**

Table 2
Impact of the augmentation on the performance of YOLO detectors on NVD

Model	Precision	Recall
YOLOv5s	56.2%	44.6%
YOLOv5s with augmented data	68.4%	49.5%
YOLOv8s	62.9%	38.4%
YOLOv8s with augmented data	74.2%	46.6%

Table 3
Impact of the augmentation on the performance of YOLO detectors on Oxford dataset (dog and cat)

Model	Precision	Recall
YOLOv5s	90%	100%
YOLOv5s with augmented data	92.4%	100%
YOLOv5m	87.2%	99%
YOLOv5m with augmented data	91.2%	99%

• Implementation Process

i. Data Collection and Preprocessing

The initial stage of the framework involves collecting and preprocessing the data. In this case study, we gather the NVD dataset and perform necessary preprocessing to ensure its quality and consistency. The dataset's images are properly organized and labeled, ready for augmentation.

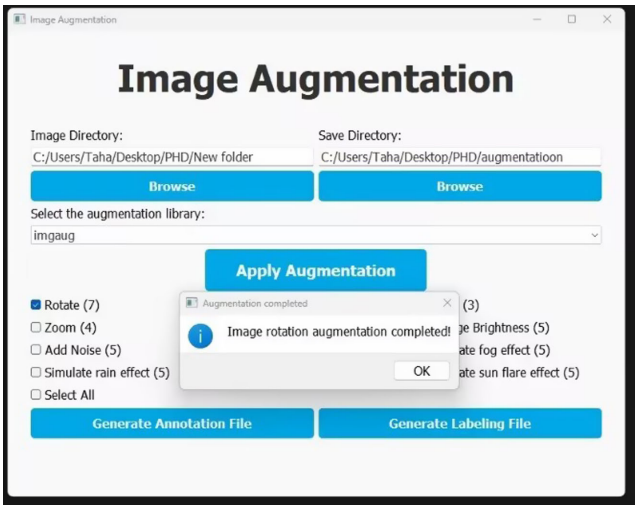
ii. Augmentation Techniques Selection

Developers or researchers can utilize the framework's GUI to choose the desired augmentation techniques suitable for the NVD dataset. For this case study, we select specific techniques that are relevant to the challenging weather conditions depicted in the dataset, such as "geometric transformation" and "weather simulation."

iii. Execution of Augmentation Techniques

After the selection of augmentation techniques, the framework automatically executes the chosen techniques on the NVD dataset. The augmentation processes generate new image samples by applying GTs like rotation, translation, and zoom, as well as WSs to mimic snow, rain, and fog. The resulting dataset includes augmented images that increase the diversity and quantity of data for ML model training.

Figure 4
Proposed GUI – geometric transformation (GT)



iv. Labeling and Classification

The framework further labels the newly augmented data with their respective classes. The data are then sorted into different folders based on these classes, enabling easy access and management during the model training process. Additionally, the framework generates labeling and annotation files compatible with STOA object detectors, facilitating model development.

3.3.2. Case study 2: Cats and dogs breeds classification Oxford dataset

A dataset featuring 37 categories of pets has been developed, comprising approximately 200 images for each category. The images within the dataset exhibit significant variations in terms of scale, pose, and lighting. Moreover, all images are accompanied by accurate ground truth annotations for breed identification, head region of interest (ROI), and pixel-level trimap segmentation (Parkhi et al., 2012).

• Implementation Process

i. Data Collection and Preprocessing

Similar to Case Study 1, the framework starts by collecting and preprocessing the Cats and Dogs Breeds Classification Oxford Dataset. The images are organized, and ground truth annotations for breed identification and head ROI are prepared.

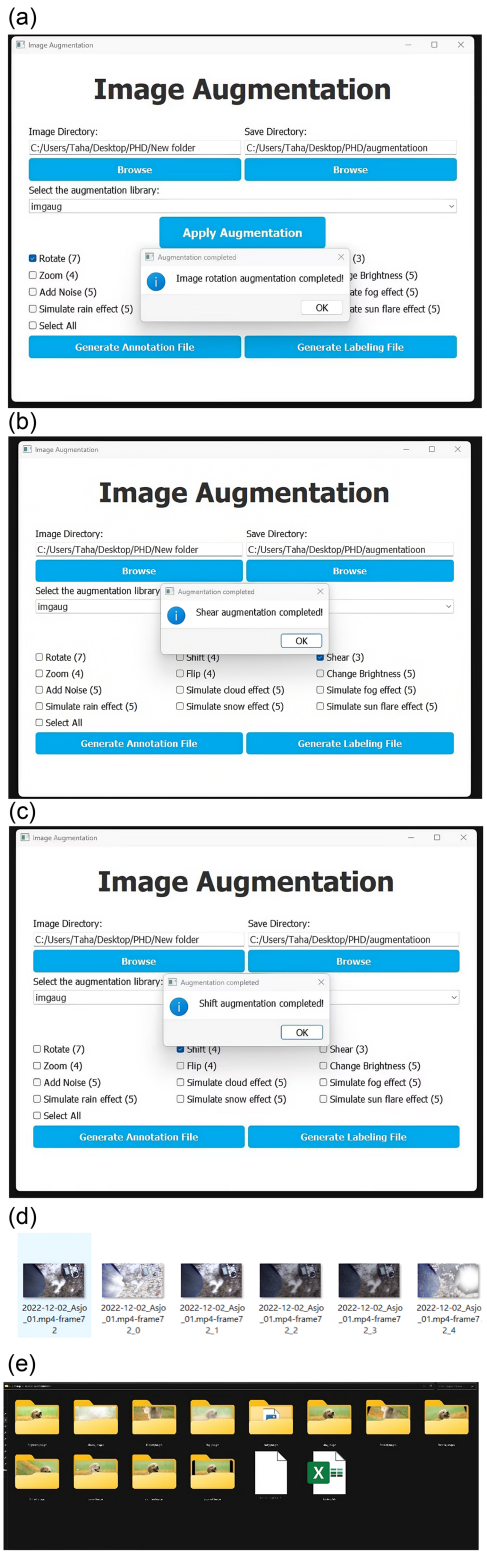
ii. Augmentation Techniques Selection

Using the framework's GUI, we select a different set of augmentation techniques relevant to this dataset to introduce variations in brightness, CS, and image filtering.

iii. Execution of Augmentation Techniques

Upon selecting the techniques, the framework applies them to the dataset, generating new samples with diverse color representations and filter effects. These augmentations enhance the dataset's variability, promoting improved generalization of ML models.

Figure 5
Sample of the generated new augmented data by applying GT



iv. Labeling and Classification

As in Case Study 1, the framework ensures that the newly augmented data are appropriately labeled and organized based on the pets' breed categories. The labeling and classification are essential for seamless integration with the training pipeline of object detection models.

Figure 6
Car detected by YOLOv5s with augmented data but YOLOv5s

Figure 7
Car detected by YOLOv8s with augmented data but YOLOv8s

3.4. Developed framework

It is a very user-friendly interface where you should only upload the dataset to the application and choose the augmentation techniques that you are planning to implement for the generated data. Additionally, the framework will provide the option to choose the desired library to use for the augmentation technique from a dropdown menu. The available options to this framework are:

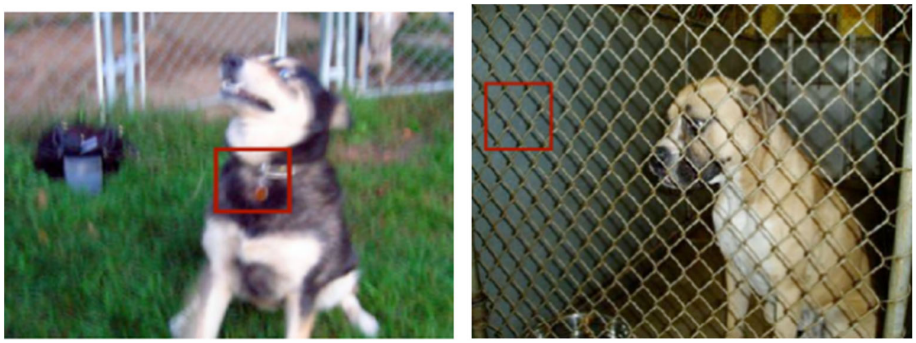
i. Keras
ii. Imgaug
iii. Albumentations
iv. Torchvision

This will provide more flexibility to the users of this framework and combining different techniques from different libraries has the potential of improving the overall outcome.

The framework will generate the following outcomes.

- Label the new augmented data with its respective labels.
- Sort the data in different files based on its classes.
- Generate a labeling file.
- Generate an annotation file that has the bounding boxes' information and is compatible with the STOA detectors.

Figure 8
Misclassification by YOLOv5 but YOLOv5m with augmented data



Figures 4 and 5 show an explanation of the different methods that can be applied using the provided tool.

4. Experimental Results

As there are different available libraries that provide the same augmentation techniques, the work starts to evaluate the time required by these libraries to execute the same augmentation technique.

The differences in performance observed among the suggested libraries justify employing a diverse set of libraries for specific functionalities as demonstrated in Table 1, ultimately improving the overall performance of the proposed tool. For instance, the torchvision library excels in processing the rotate augmentation, while imgaug proves to be the most efficient in handling RE features. Additionally, albumentations demonstrate superior performance in augmentation methods involving color changes, such as greyscale conversion.

The performance of STOA detectors is evaluated via Table 2 to determine the impact of augmented data. Through experimental testing on the YOLO detector using various datasets, the necessity of incorporating such a tool into the DM and preparation stage is substantiated.

Despite the STOA detectors' limited effectiveness in dealing with the demanding scenario presented in the NVD dataset, a notable improvement in both precision and recall is evident for both YOLOv5 and YOLOv8 as summarized in Table 3. This enhancement is attributed to the utilization of the tool's proposed augmented data during the implementation process. The enhancement of the recall was 12% on both detectors, while around 7% for the recall. Objects detection for NVD data are shown in Figures 6 and 7.

To confirm the universality of the proposed tool, we also assessed its effect on a widely used dataset, namely the "Oxford (dog and cat)." The evaluation reveals that the precision accuracy registers an improvement of approximately 3–4% following the generation of augmented data using the tool as in Figure 8.

5. Conclusion

The scarcity of data poses a significant challenge in the development of ML applications on a daily basis. With many sectors keeping their data confidential, obtaining the necessary volume of data for constructing accurate ML models has become increasingly difficult for developers and researchers. In this paper, we presented a comprehensive framework to support the DM stage of the ML development life cycle, with a specific focus on data augmentation techniques.

Through the proposed framework, developers and researchers can efficiently implement various augmentation methods without the need for manual coding. By automating the augmentation process, the framework reduces the time and effort required for data labeling and classification into respective folders, ultimately enhancing the overall DM process. Our two implementation case studies on diverse datasets, the NVD and the Cats and Dogs Breeds Classification Oxford Dataset, demonstrated the practical effectiveness of the framework in improving ML model performance for object detection tasks.

However, this study has certain limitations that should be acknowledged. Firstly, while the proposed framework streamlines the DM stage, it relies on the selection and combination of specific augmentation techniques. There may be instances where alternative or additional techniques could further improve data quality and model performance. Future research could explore an adaptive or automated method for selecting the most relevant augmentation techniques based on the specific characteristics of the dataset.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Supportive Materials

The tool can be accessed using the following link: https://github.com/TahaKh99/Image_Augmentor.

References

Adewumi, T., Alkhaled, L., Mokayed, H., Liwicki, F., & Liwicki, M. (2022). ML_LTU at SemEval-2022 task 4: T5 towards identifying patronizing and condescending language. In *Proceedings of the 16th International Workshop on Semantic Evaluation*, 473–478. https://doi.org/10.18653/v1/2022.semeval-1.64

Alkhaled, L., Adewumi, T., & Sabry, S. S. (2023). Bipol: A novel multi-axes bias evaluation metric with explainability for NLP. *Natural Language Processing Journal, 4*, 100030. https://doi.org/10.1016/j.nlp.2023.100030

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., . . . , & Zimmermann, T. (2019). Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice*, 291–300. https://doi.org/10.1109/ICSE-SEIP.2019.00042

Ashmore, R., Calinescu, R., & Paterson, C. (2021). Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *ACM Computing Surveys*, *54*(5), 111. https://doi.org/10.1145/3453444

Baylor, D., Breck, E., Cheng, H. T., Fiedel, N., Foo, C. Y., Haque, Z., . . . , & Zinkevich, M. (2017). TFX: A tensorflow-based production-scale machine learning platform. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 1387–1395. https://doi.org/10.1145/3097983.3098021

Borges, O., Couto, J., Ruiz, D., & Prikladnicki, R. (2021). Challenges in using machine learning to support software engineering. In *Proceedings of the 23rd International Conference on Enterprise Information Systems*, *2*, 224–231.

Brown, S. (2021). *Machine learning, explained*. Retrieved from: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and flexible image augmentations. *Information*, *11*(2), 125. https://doi.org/10.3390/info11020125

Daisy, D. (2021). *Machine learning vs. Normal programming: What's the difference?* Retrieved from: https://datasciencenerd.com/machine-learning-vs-normal-programming-what-is-the-difference/

de Souza Nascimento, E., Ahmed, I., Oliveira, E., Palheta, M. P., Steinmacher, I., & Conte, T. (2019). Understanding development process of machine learning systems: Challenges and solutions. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 1–6. https://doi.org/10.1109/ESEM.2019.8870157

Douglass, M. J. J. (2020). Book review: Hands-on machine learning with Scikit-learn, Keras, and TensorFlow. *Physical and Engineering Sciences in Medicine*, *43*(3), 1135–1136. https://doi.org/10.1007/s13246-020-00913-z

Ilyas, I. F., & Chu, X. (2019). *Data cleaning*. USA: Morgan & Claypool Publishers.

Jung, A. B., Wada, K., Crall, J., Tanaka, S., Graving, J., Reinders, C., . . . , & Laporte, M. (2020). *Imgaug*. USA: GitHub.

Kanchi, S., Pagani, A., Mokayed, H., Liwicki, M., Stricker, D., & Afzal, M. Z. (2022). EmmDocClassifier: Efficient multimodal document image classifier for scarce data. *Applied Sciences*, *12*(3), 1457. https://doi.org/10.3390/app12031457

Khan, M. A. U., Nazir, D., Pagani, A., Mokayed, H., Liwicki, M., Stricker, D., & Afzal, M. Z. (2022). A comprehensive survey of depth completion approaches. *Sensors*, *22*(18), 6969. https://doi.org/10.3390/s22186969

Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2018). Data scientists in software teams: State of the art and challenges. *IEEE Transactions on Software Engineering*, *44*(11), 1024–1038. https://doi.org/10.1109/TSE.2017.2754374

Machap, K., & Khamis, T. (2022). Assessing tools to analyze the techniques and mechanism for network risk minimization. *Journal of Applied Technology and Innovation*, *6*(1), 14–17.

Mikolajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. In *2018 International Interdisciplinary PhD Workshop*, 117–122. https://doi.org/10.1109/IIPHDW.2018.8388338

Mokayed, H., Nayebiastaneh, A., De, K., Sozos, S., Hagner, O., & Backe, B. (2023). Nordic Vehicle Dataset (NVD): Performance of vehicle detectors using newly captured NVD from UAV in different snowy weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5314–5322.

Morgunov, A., (2023). *The life cycle of a machine learning project: What are the stages?* Retrieved from: https://neptune.ai/blog/life-cycle-of-a-machine-learning-project

Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. (2012). Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3498–3505. https://doi.org/10.1109/CVPR.2012.6248092

Roh, Y., Heo, G., & Whang, S. E. (2021). A survey on data collection for machine learning: A big data-AI integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, *33*(4), 1328–1347. https://doi.org/10.1109/TKDE.2019.2946162

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3234–3243.

Sheng, V. S., & Zhang, J. (2019). Machine learning with crowdsourcing: A brief summary of the past research and future directions. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 9837–9843. https://doi.org/10.1609/aaai.v33i01.33019837

Taddeo, M., McCutcheon, T., & Floridi, L. (2019). Trusting artificial intelligence in cybersecurity is a double-edged sword. *Nature Machine Intelligence*, *1*(12), 557–560. https://doi.org/10.1038/s42256-019-0109-1

Voon, W., Hum, Y. C., Tee, Y. K., Yap, W. S., Salim, M. I. M., Tan, T. S., . . . , & Lai, K. W. (2022). Performance analysis of seven Convolutional Neural Networks (CNNs) with transfer learning for Invasive Ductal Carcinoma (IDC) grading in breast histopathological images. *Scientific Reports*, *12*(1), 19200. https://doi.org/10.1038/s41598-022-21848-3

Whang, S. E., & Lee, J. G. (2020). Data collection and quality challenges for deep learning. *Proceedings of the VLDB Endowment*, *13*(12), 3429–3432. https://doi.org/10.14778/3415478.3415562

Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding data augmentation for classification: When to warp? In *2016 International Conference on Digital Image Computing: Techniques and Applications*, 1–6. https://doi.org/10.1109/DICTA.2016.7797091

Zero, J., & Hug, N. (2016). *Torchvision*. Retrieved from: https://github.com/pytorch/vision

How to Cite: Alkhaled, L., & Khamis, T. (2024). Supportive Environment for Better Data Management Stage in the Cycle of ML Process. *Artificial Intelligence and Applications*, *2*(2), 121–128. https://doi.org/10.47852/bonviewAIA32021224