Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Data Augmentation using Generative-AI

Samarth R Gowda[1], Pavithra H C.[2], Sunitha R.[3], Somaiah K M.[4], Suraj S H.[5], Yashas R Rao[6]

[1,3,4,5,6]Student, [2]Faculty, Artificial Intelligence and Machine Learning Department, BNMIT, Bengaluru, India.

E-mail: [1]samarthrgowda2002@gmail.com, [2]pavithrahc@bnmit.in, [3]sunithar@bnmit.in, [4]dhanushkm11@gmail.com, [5]surajsh0115@gmail.com, [6]yashasrao2002@gmail.com

Abstract

This study presents an approachable tool for data augmentation that makes use of artificial intelligence (AI). It can handle text and visual data, assisting customers in optimizing their data collecting for a range of applications. The system breaks down CSV documents providing insights using libraries such as transformers, which are used in the field of Natural Language Processing (NLP). It assesses the insights in addition to applying data augmentation techniques like word control and equivalent substitution. This method improves the text data by quickly balancing the final dataset. This study uses Generative III-disposed Organizations (GANs) to handle the images. Users can change settings like rotation, scale, and translation for a variety of high-quality images. This use case goes beyond simple growth and touches on the territory of artificial intelligence. With an emphasis on usability, the User Interface (UI) enables researchers to customize the processes according to their specific datasets.

Keywords: NLP, Image Processing, Equivalent Substitution, Word Control, GAN, User Interface (UI)

1. Introduction

The need to make data augmentation in machine learning more accessible and faster prompted this effort. Today's tools typically specialize in natural language processing or image augmentation and are standalone. This presents a challenge for researchers and practitioners who work with datasets that have both textual and visual features. Our research develops a simple to use user interface that integrates natural language processing and image augmentation

DOI: https://doi.org/10.36548/jiip.3.005
Received: 10.06.2024, received in revised form: 06.07.2024, accepted: 11.07.2024, published: 24.07.2024

smoothly to address this difficulty. Textual data (such as reviews and social media postings) can be precisely enhanced by using transformers to execute difficult natural language processing (NLP) operations like sentiment analysis and topic modelling. The platform also gives users the ability to enhance images using Generative Adversarial Networks (GANs). Customizing parameters such as rotation, scaling, and translation enables users to produce a wide range of excellent images that closely mimic the original data distribution. This goes beyond simple augmentation and into the field of data synthesis, which is especially useful for datasets with small sample sizes. A versatile tool that can handle a wide range of datasets from different domains—like sentiment analysis of product reviews with images, medical diagnosis using X-rays and patient reports, or autonomous vehicle training using traffic signs and sensor data—is ensured by this unified approach. The goal of the research is to improve model robustness and generalization by enhancing the datasets with these methods, which will result in machine learning applications that are more accurate and dependable. Additionally, the research gives ethical issues in data augmentation top priority. We prioritize user privacy by making sure that information is safe on the platform. The user-friendly interface developed is well-defined with options and parameters, enabling users to make well-informed decisions regarding their augmentation strategy. By encouraging the moral and approachable application of generative AI techniques for dataset enhancement.

2. Related Work

This study discusses an innovative approach to augmenting sentiment analysis datasets using the generative capabilities of large language models like ChatGPT. By leveraging ChatGPT, the authors explore methods to generate synthetic data that can be used to supplement existing datasets, ultimately aiming to enhance the performance of smaller machine learning models without the need for extensive data collection and manual annotation. It focuses on two main questions: whether ChatGPT-augmented data can boost the performance of smaller models to a level comparable with larger, more resource-intensive models, and whether the benefits of using augmented data are consistent across different model scales [1].

An in-depth review of Generative Adversarial Networks (GANs) and their application in medical image analysis, focuses on their role in addressing data insufficiency and class imbalance in medical image datasets. By using GANs, researchers have been able to generate

augmented medical images, thereby increasing the size of datasets and achieving more balanced class distributions, which are crucial for effective classification and segmentation tasks in medical diagnosis [2].

The critical role of data augmentation in medical image analysis, emphasizing its importance in improving the robustness and generalization of models is explored. It highlights the application of Generative Adversarial Networks (GANs) in generating realistic medical images to address common challenges like insufficient data and imbalanced class distributions in medical image datasets [3].

This study [4] provides a literature review on the application of Generative Adversarial Networks (GANs) in the field of ophthalmology image domains, highlighting their ability to improve diagnostic capabilities through image synthesis and image-to-image translation. GANs, composed of a generator and a discriminator, have gained attention for their performance in these areas, yet their adoption in ophthalmology has been limited. The review aims to discuss the contributions of GANs in this domain and identify potential future research directions.

The research [5] introduces a novel data augmentation method based on Generative Adversarial Network with Mixed Attention Mechanism (MA-GAN), designed to address common issues in GANs, such as unstable training and low-quality generated images. The proposed approach aims to generate consistent objects or scenes by correlating remote features in the image, thereby enhancing the ability to create detailed imagery.

The significant challenge in data augmentation: the lack of specific rules or guidelines for determining the optimal number of synthetic data samples needed to maximize deep learning-based diagnostic performance is examined in this research. To tackle this issue, the authors propose an automation pipeline that uses generative adversarial networks (GANs) to systematically find the best multiple of data augmentation for achieving optimal diagnostic performance with limited datasets [6].

The framework based on Generative Adversarial Networks (GANs) to create synthetic structural brain networks in Multiple Sclerosis (MS), addressing the challenge of limited and imbalanced datasets in the biomedical domain is presented in this research [7]. Machine learning frameworks have shown potential in handling complex data structures, achieving notable results in areas like brain imaging. However, training these models requires a large

collection of data, which can be difficult to obtain due to acquisition costs, accessibility, and pathology-related variability.

This study discusses the development of an innovative data augmentation technique for improving the performance of Convolutional Neural Networks (CNNs) in the context of limited and imbalanced datasets, particularly in medical imaging. The proposed model, Inception-Augmentation GAN (IAGAN), aims to address the limitations of traditional data augmentation methods and the challenges associated with using Generative Adversarial Networks (GANs) for augmenting training data [8].

The study [9] explores the use of Generative Adversarial Networks (GANs) as a data augmentation method to enhance machine fault detection, addressing the challenge of limited and imbalanced datasets in the early phases of predictive maintenance. Predictive maintenance in industrial factories has gained traction with the increased use of the Internet of Things (IoT) and artificial intelligence (AI) algorithms for data management. However, limited machine fault samples and imbalanced data complicate fault classification training. This study proposes GAN-based data augmentation to enhance the dataset, leading to improved accuracy in the fault detection model's training process.

The work [10] explores the limitations of pre-trained BERT-style models, such as RoBERTa, ALBERT, and XLNet, which are based on the Transformer architecture, in the context of certain Natural Language Processing (NLP) tasks. Despite their success in revolutionizing NLP and pushing the state of the art, these models have inherent limitations that affect their ability to process specific types of information or data structures. This research identifies limitations in BERT-style models for sequence tasks, proposing practical enhancements.

The research explores practical and scalable text data augmentation techniques to address the challenge of limited data for training deep neural networks. The "Big Data Wall" is a significant issue, particularly for minority language communities, smaller organizations, and laboratories that cannot compete with tech giants like Google, Amazon, Facebook, Apple, and Microsoft. The study emphasizes using robust, scalable, and easy-to-implement data augmentation pre-processing techniques similar to those used in computer vision to overcome this limitation [11].

The research delves into the concept of Data Augmentation (DA) in the context of Machine Learning (ML) and Deep Learning (DL), emphasizing its role in reducing overfitting and enhancing the generalization power of ML models. The research provides an overview of various DA techniques, their applications in different domains, and the theoretical principles underlying these techniques [12].

The study proposes a novel approach to data augmentation for plant phenotyping problems, specifically leaf segmentation and leaf counting, using conditional Generative Adversarial Networks (cGANs). The objective is to address the problem of data scarcity by generating artificial images that resemble realistic plant images, allowing for improved training and generalization of machine learning models in these tasks [13].

Various Generative AI (ChatGPT) enhances text classification, experiments reveal significant macro F1 improvement and notably in the BERT model is discussed. BERT model macro F1 scores increase significantly, 30 diverse samples outperform 6 duplicates, and longer texts contribute to higher accuracy scores [14].

The AugGPT, leveraging large language models presented in this study, enhances few-shot learning in NLP via text data augmentation. Outperforming existing methods, it produces diverse, faithful samples, overcoming data scarcity. AugGPT outperforms other methods, demonstrating superior performance in testing accuracy and distribution of augmented samples for few-shot learning text classification tasks [15].

3. Proposed Work

The data flow diagram in Figure 3.1 shows the steps that data goes through in the generative AI data augmentation research: from user interaction with a front-end interface to data processing and augmentation, and finally to email distribution. It demonstrates the uploading, storing, processing, and returning of CSV and image files to the user following augmentation. In addition, the diagram shows the relationships between the internal processes that control data flow and how the system functions as a whole.

- **Front-End Interface:** Allows users to upload CSV or image files for augmentation. Each file type is covered in its own section, with controls for choosing files from the user's system. A text field is provided for users to enter their email addresses to receive the augmented data. A button initiates the data augmentation process. When clicked,

the email address and uploaded files are sent to the backend for processing. The system gives users visual feedback, such as displaying error messages or indicating a successful file upload.

- **AWS S3 Storage:** Stores the uploaded CSV and image files for further processing. This initial storage step ensures the files are accessible for augmentation. Stores the data produced by the augmentation processes, ensuring that it can be retrieved for further use or distribution. Allows processes to fetch data from S3 for processing or email attachment.

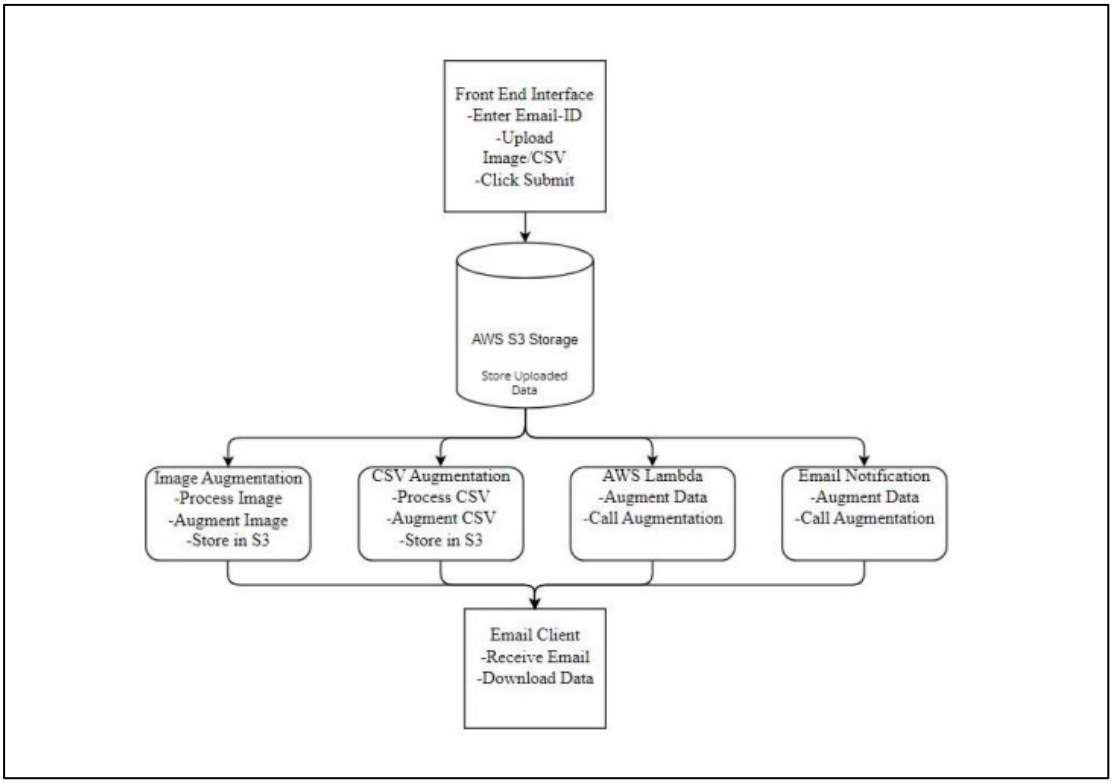

Figure 3.1. Block diagram

- **Data Augmentation Processes:** Processes image data using Generative Adversarial Networks (GANs), creating artificial images based on specific requirements (e.g., masks, labels). The enhanced images are then saved in S3. Applies augmentation methods to CSV data, including conditional generation, oversampling, and data synthesis, handling CSV files and saving the enhanced data to S3. Runs data

augmentation code using serverless computing, providing scalability and flexibility by initiating different functions based on the type of data (image or CSV)

- **Email Notification:** The Email Notification component is responsible for sending augmented data to users through email. Creates email messages containing download links for the augmented data. This step ensures proper formatting and inclusion of relevant information. Uses an email service like AWS Simple Email Service to send the composed email to the user's provided email address. This step involves the actual email delivery process.

- **Email Client:** The Email Client represents the user's email system where the augmented data is received. It is the final step in the data flow, where the user downloads the augmented CSV or image files.

4. System Implementation

4.1 Dataset

The research utilizes datasets similar to popular text and image collections, where the data originates from users. This user-provided data, can be in the form of text and images. An example of a text dataset format is "Text": "label", where each text snippet has a corresponding label indicating its category or sentiment. This approach is commonly used for text classification tasks, allowing models to learn the relationships between text and labels. For our experiments, we specifically tested models on the MNIST dataset for tasks like Generative Adversarial Networks (GANs). The MNIST database (Modified National Institute of Standards and Technology database) is a large collection of handwritten digits.

4.2 Model Description

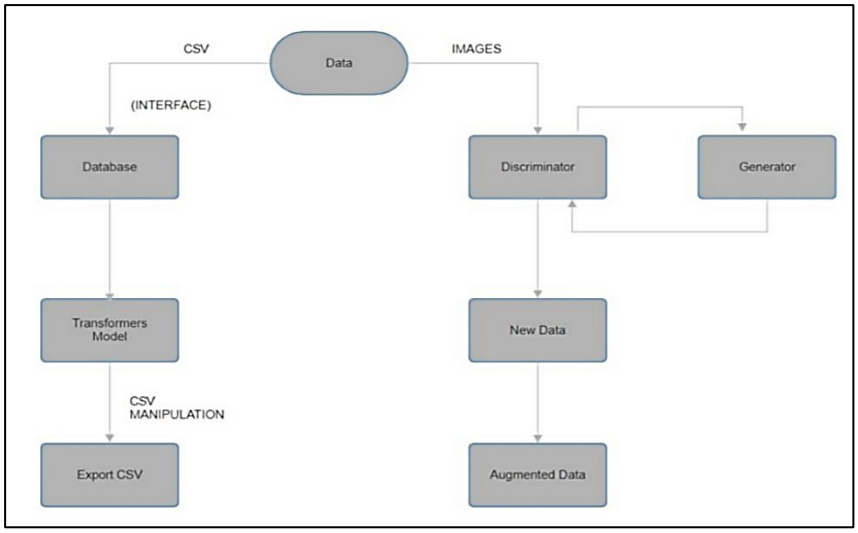

Figure 4.1. Architecture of the Model

Figure 4.1 shows the architecture of the model which has the following modules:

1. Front-End Interface Module: The Front-End Interface Module represents the user-facing part of the system where users interact with the application. It allows users to upload CSV or image files and includes feedback mechanisms to inform users of successful uploads or errors. A text field is provided for users to enter their email addresses, where they will receive the augmented data. There is a trigger that initiates the data processing workflow when clicked, sending the uploaded files and email address to the back-end for processing. The module provides a user-friendly interface with clear instructions and visual feedback to guide users through the data augmentation process. For the frontend application, we use React, which allows us to build user interfaces from individual components. We have created our own React components, such as Thumbnail and Video, and combined them into complete screens, pages, and applications to perform the required actions.

- File Upload: Users upload CSV or image files through the front-end interface. The interface has separate sections for each type of data (CSV and images). The system validates the uploaded files to ensure they meet the expected format (e.g., CSV files have a valid structure, images are in standard formats like JPEG or PNG).

- Email Input: Users enter their email address where the augmented data will be sent. The system checks the email address format to ensure it's valid.
- Submit Action: When the submit button is clicked, the system triggers the data processing phase, sending the uploaded files and the email address to the AWS S3 storage for further processing.

2. Data Storage Module: The Data Storage Module manages the storage of uploaded files and augmented data. It uses AWS S3 for scalable and persistent storage, providing flexibility and reliability. It stores the raw CSV or image files uploaded by users in AWS S3, as well as the data produced by the augmentation processes, ensuring that it is accessible for further processing or email notification. The module structures the data in S3 to allow for easy retrieval and management, maintaining clear organization between raw and augmented data.
3. Data Augmentation Module: The Data Augmentation Module handles the core logic for augmenting CSV and image data. It includes various techniques to increase data diversity and improve data augmentation outcomes. Processing CSV data with appropriate augmentation techniques, such as data synthesis, oversampling, or conditional generation. The augmented data is then stored in AWS S3. For image data, it uses Generative Adversarial Networks (GANs) or other augmentation techniques to generate synthetic images, which are also stored in AWS S3. The module condenses spatial information into a single feature vector to help the model focus on key attributes without unnecessary data volume.
4. Text Preprocessing Module: The Text Preprocessing Module deals with processing text data to prepare it for augmentation. This module handles operations that transform raw text into a suitable format for further processing. Removing unwanted characters, punctuation, or symbols from text data to ensure consistency. Converting text to a normalized format, such as lowercasing or removing stopwords. Breaking text into tokens (individual words or sub-words) for processing.
5. Serverless Computing Module: The Serverless Computing Module uses AWS Lambda to execute serverless code for data augmentation processes. This approach provides scalability and flexibility to handle varying workloads. It uses specific serverless functions to perform data augmentation for both CSV and image data, managing code execution efficiently and ensuring optimal resource utilization.

6. Email Notification Module: The Email Notification Module manages sending email notifications to users after data augmentation is complete. It uses AWS SES for email delivery, creating messages with download links for the augmented data. The module ensures proper formatting and includes all necessary information for users to access their data. It sends the composed emails to the user's provided address, tracks delivery, and handles potential issues. Mechanisms are also implemented to address problems with email delivery, such as incorrect email addresses or delivery failures.
7. Model Development

1. Model Loading:

   The Pegasus model architecture is used for conditional text generation, utilizing PegasusTokenizerFast and PegasusForConditionalGeneration from the transformers library.
2. Model Initialization:

   It initialises the model by providing the model name and the directory to save/load the model in. If the model hasn't been downloaded before, it is downloaded and stored to the specified directory.
3. Tokenizer Initialization:

   Similarly, the code initializes the tokenizer by specifying the directory to save/load the tokenizer. If the tokenizer is not already downloaded, it downloads and saves the tokenizer to the specified directory using PegasusTokenizerFast.from_pretrained (model_name).
4. Paraphrasing Function:

   The get_paraphrased_sentences function takes the initialized model and tokenizer along with a sentence as input and generates paraphrased sentences. It tokenizes the input sentence, generates paraphrased sentences using the model, and decodes the generated sentences using the tokenizer to return them as text.
5. Generator Model (make_generator_model):

   This function defines a generator model using TensorFlow's Keras Sequential API. It consists of several layers, including dense, batch normalization, leaky ReLU, and transpose convolutional layers. The generator takes a 100-dimensional random noise vector as input and produces a 28x28x1 image as output.
6. Discriminator Model (make_discriminator_model):

This function defines a discriminator model using TensorFlow's Keras Sequential API. It consists of several convolutional layers, followed by leaky ReLU activation functions and dropout layers. The discriminator takes a 28x28x1 image as input and outputs a single scalar value representing the probability that the input image is real.

7. Model Initialization:

   The discriminator model is instantiated using make_discriminator_model. A sample noise vector is generated and passed through the discriminator to obtain a decision (decision) on the generated image.

8. discriminator_loss:

   Computes the adversarial loss for the discriminator by comparing the real and fake output distributions using cross-entropy loss.

9. generator_loss:

   Computes the adversarial loss for the generator by maximizing the probability that the generated images are classified as real.

10. Optimizers and Checkpoints:

    Adam optimizers are instantiated for both the generator and discriminator models. Checkpointing is set up to save and restore the model weights during training.

11. Training Step (train_step):

    This function performs one training step on a batch of real images. It computes the generator and discriminator losses using the generated and real images, respectively, and applies gradient updates to the model parameters.

12. Training Loop (train):

    This loop iterates over the dataset for a specified number of epochs. For each epoch, it calls train_step on each batch of images in the dataset and saves generated images at regular intervals. It also saves model checkpoints every 15 epochs.

13. Image Generation and Saving (generate_and_save_images):

    This function generates and saves images produced by the generator model. It takes the generator model, epoch number, and a test input noise vector as input and saves the generated images in a grid format.

5. Results and Discussion

The data augmentation process begins with handling CSV and image data using advanced techniques and platforms. For CSV data, techniques such as data synthesis,

oversampling, and conditional generation are employed to enhance data diversity and improve outcomes. The scikit-learn library is used for data manipulation, and the augmentation processes are executed on the TensorFlow platform. The augmented data is then stored in AWS S3, which provides scalable and reliable cloud storage solutions. For image data, Generative Adversarial Networks (GANs) or other image augmentation methods are utilized to generate synthetic images. The Keras library facilitates this augmentation, with processed images also being saved in AWS S3 for persistent storage.

Text preprocessing involves transforming raw text into a format suitable for augmentation. This includes removing unwanted characters, normalizing text (e.g., lowercasing, stopword removal), and breaking text into tokens. Python libraries such as NLTK handle these preprocessing tasks. For more advanced text processing and embedding, the TensorFlow platform is used, along with the Hugging Face transformers library. Processed text data is stored in AWS S3, ensuring both accessibility and organization. To integrate these processes, the AWS Lambda serverless computing platform is used to execute the data augmentation and preprocessing workflows efficiently, with APIs facilitating seamless interaction between different system component

The results and discussions from the data augmentation research using generative AI revealed several key insights. By successfully increasing the volume and diversity of datasets for both CSV and image data, the augmentation procedure expanded the amount of training data available to machine learning models. Testing showed that all essential components of the system operated as intended, integrating well and facilitating a seamless end-to-end workflow. Validation confirmed that the augmented data was accurate and of high quality, adhering to expected patterns and benchmarks. However, several challenges and limitations were noted, including managing edge cases and ensuring consistent data quality during the augmentation process.

Overall, the experiment demonstrated the effectiveness of generative AI in data augmentation, balancing recall and precision. The findings provide a strong foundation for further development and research, advancing our understanding of data augmentation in machine learning contexts.

Figures 5.1 and 5.2 shows the user interface home page and the image upload page, respectively.

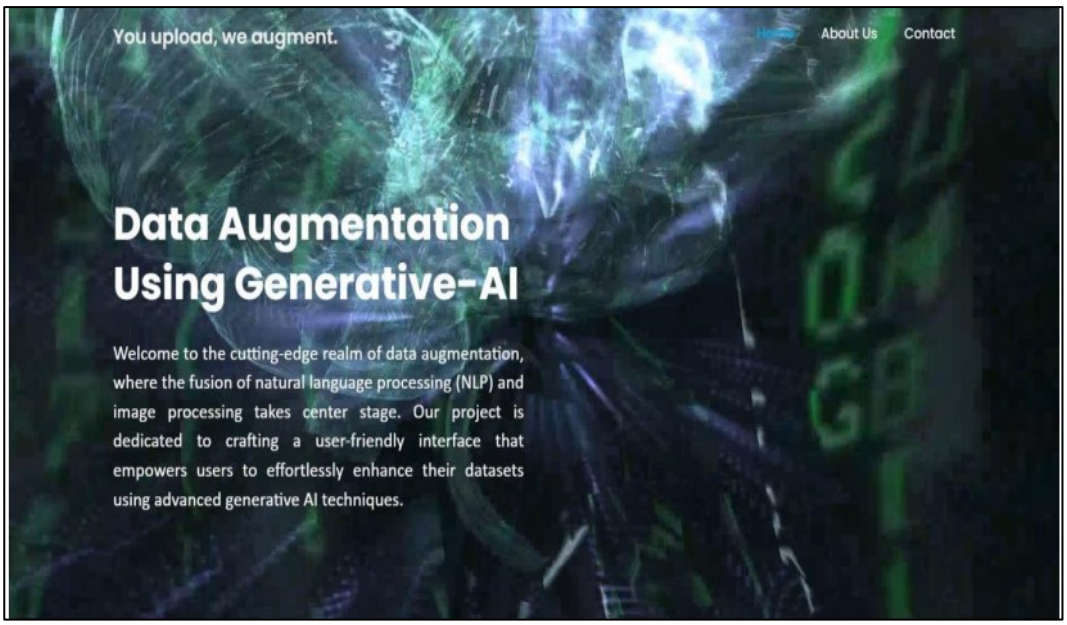

Figure 5.1. Home Page

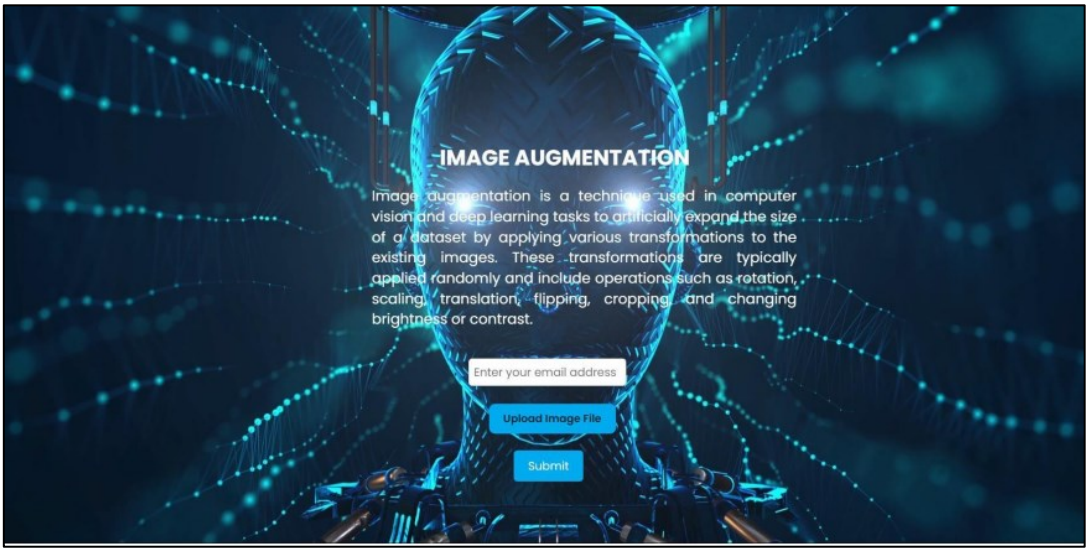

Figure 5.2. Image Upload Page

The Figure 5.3 shows the results of CSV augmentation.

- **CSV Data Augmentation:** The data augmentation process for CSV files increased the dataset's diversity and volume. The augmented CSV files contained synthesized records that added value to the original dataset, enabling new variations for model training and testing. Key metrics to evaluate CSV augmentation included the number of records added and the overall increase in data diversity.

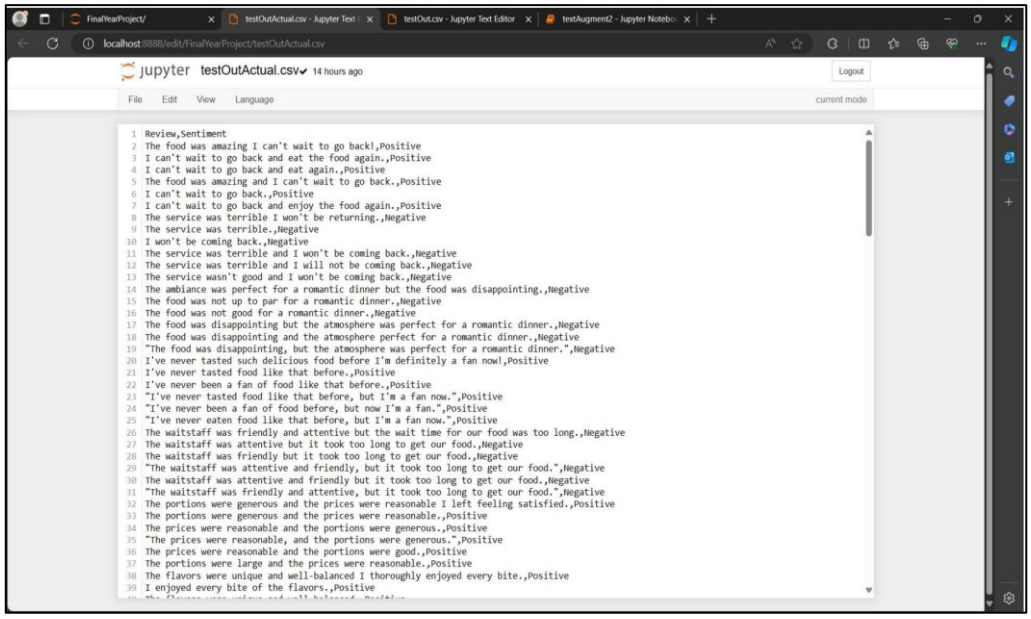

Figure.5.3. Results of CSV Augmentation

- **Image Data Augmentation:** The image augmentation process, using Generative Adversarial Networks (GANs), successfully generated synthetic images. These images reflected a range of visual characteristics that enhanced the original dataset's variability. Key metrics to evaluate image augmentation included the number of synthetic images generated and their similarity to the original data. Figure 5.4 shows the results of image augmentation.

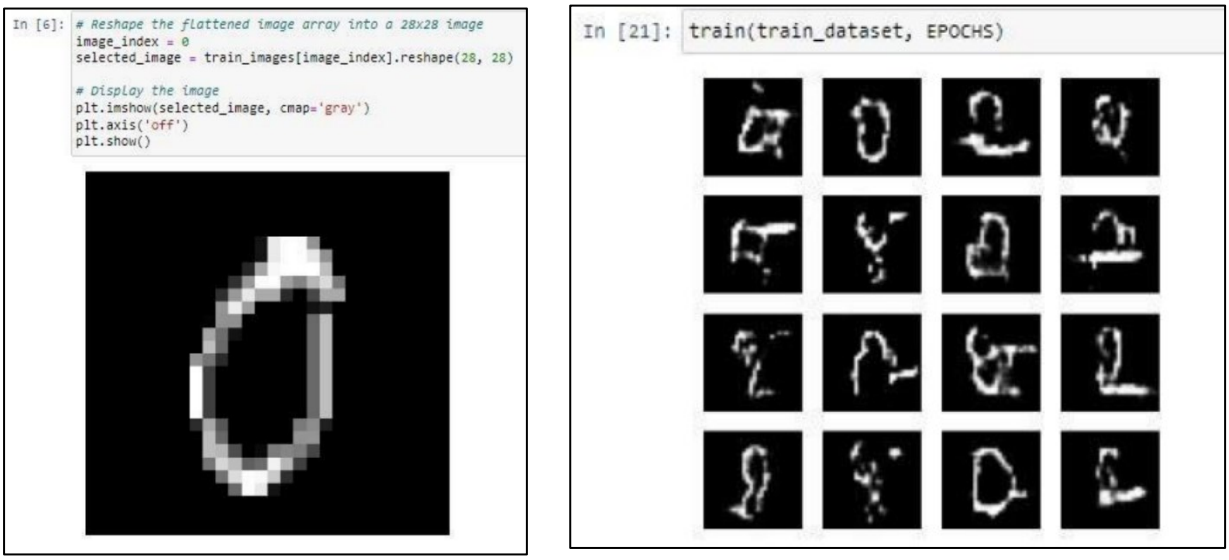

Figure 5.4. Results of Image Augmentation

6. Conclusion

The primary goal of the generative AI data augmentation research was achieved: increasing the diversity and volume of datasets led to better performance in machine learning tasks. The research focused on creating a system that allowed users to input CSV and image files for enhancement using advanced techniques like Generative Adversarial Networks (GANs). After the data was validated, users received the updated data through email. Throughout the research, the system reliably arranged and enhanced data. Testing and validation improved the model's performance and confirmed that the augmented data met quality standards. The research successfully addressed overfitting issues and improved the models' ability to generalize by providing additional training data. Although there were challenges such as handling unpredictable conditions, ensuring consistent data quality, and addressing potential performance issues, careful testing and validation minimized these problems, resulting in a dependable and effective data augmentation system.

References

[1] Woźniak, Stanisław, and Jan Kocoń. "From Big to Small Without Losing It All: Text Augmentation with ChatGPT for Efficient Sentiment Analysis." In 2023 IEEE International Conference on Data Mining Workshops (ICDMW), Shanghai, China. IEEE, 2023. 799-808.

[2] Makhlouf, Ahmed, Marina Maayah, Nada Abughanam, and Cagatay Catal. "The use of generative adversarial networks in medical image augmentation." Neural Computing and Applications 35, no. 34 (2023): 24055-24068.

[3] Biswas, Angona, Nasim Md Abdullah Al, Al Imran, Anika Tabassum Sejuty, Fabliha Fairooz, Sai Puppala, and Sajedul Talukder. "Generative adversarial networks for data augmentation." In Data Driven Approaches on Medical Imaging, Cham: Springer Nature Switzerland, 2023. pp. 159-177.

[4] You, Aram, Jin Kuk Kim, Ik Hee Ryu, and Tae Keun Yoo. "Application of generative adversarial networks (GAN) for ophthalmology image domains: a survey." Eye and Vision 9, no. 1 (2022): 6.

[5] Yang, Yu, Lei Sun, Xiuqing Mao, and Min Zhao. "Data augmentation based on generative adversarial network with mixed attention mechanism." Electronics 11, no. 11 (2022): 1718.

[6] Kong, Hyoun-Joong, Jin Youp Kim, Hye-Min Moon, Hae Chan Park, Jeong-Whun Kim, Ruth Lim, Jonghye Woo, Georges El Fakhri, Dae Woo Kim, and Sungwan Kim. "Automation of generative adversarial network-based synthetic data-augmentation for maximizing the diagnostic performance with paranasal imaging." Scientific Reports 12, no. 1 (2022): 18118.

[7] Barile, Berardino, Aldo Marzullo, Claudio Stamile, Françoise Durand-Dubief, and Dominique Sappey-Marinier. "Data augmentation using generative adversarial neural networks on brain structural connectivity in multiple sclerosis." Computer methods and programs in biomedicine 206 (2021): 106113.

[8] Motamed, Saman, Patrik Rogalla, and Farzad Khalvati. "Data augmentation using Generative Adversarial Networks (GANs) for GAN-based detection of Pneumonia and COVID-19 in chest X-ray images." Informatics in medicine unlocked 27 (2021): 100779.

[9] Bui, Van, Tung Lam Pham, Huy Nguyen, and Yeong Min Jang. "Data augmentation using generative adversarial network for automatic machine fault detection based on vibration signals." Applied Sciences 11, no. 5 (2021): 2166.

[10] Chernyavskiy, Anton, Dmitry Ilvovsky, and Preslav Nakov. "Transformers:“the end of history” for natural language processing?." In Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21, Springer International Publishing, 2021. 677-693.

[11] Coulombe, Claude. "Text data augmentation made simple by leveraging nlp cloud apis." arXiv preprint arXiv:1812.04718 (2018).

[12] Pellicer, Lucas Francisco Amaral Orosco, Taynan Maier Ferreira, and Anna Helena Reali Costa. "Data augmentation techniques in natural language processing." Applied Soft Computing 132 (2023): 109803.

[13] Zhu, Yezi, Marc Aoun, Marcel Krijn, Joaquin Vanschoren, and High Tech Campus. "Data Augmentation using Conditional Generative Adversarial Networks for Leaf Counting in Arabidopsis Plants." In BMVC, vol. 2018, 121-125.

[14] Zhao, Huanhuan, Haihua Chen, and Hong-Jun Yoon. "Enhancing Text Classification Models with Generative AI-aided Data Augmentation." In 2023 IEEE International Conference On Artificial Intelligence Testing (AITest), Athens, Greece IEEE, 2023. 138-145.

[15] Dai, Haixing, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao. "Auggpt: Leveraging chatgpt for text data augmentation." arXiv preprint arXiv:2302.13007 (2023).