Improving CNN Performance through Generative AI: A Data Augmentation Approach

Bryan Kyritz
Electrical and Computer Engineering
Stevens Institute of Technology
bkyritz@stevens.edu

***Abstract*—This project investigates the impact of generative AI on the performance of convolutional neural networks (CNNs) in image classification tasks, specifically in the context of limited data. Three distinct CNN models were implemented and compared, each trained on a unique dataset: the original images of cats and dogs, the same images augmented by OpenAI's DALL-E, and a further enhanced dataset with neural style transfer applied. The findings highlight the potential of using generative AI for data augmentation. The model trained on the most augmented dataset achieved the highest accuracy of 90.86% on the test set, which was a 3.16% increase over the model trained on the original dataset. The study underscores the value of integrating state-of-the-art AI methods to improve the generalizability and performance of CNNs, even in data-limited scenarios.**

I. Introduction

As the field of artificial intelligence continues to evolve, generative diffusion models such as Midjourney, DALL-E, and Stable Diffusion are gaining recognition for their ability to generate content that mirrors human creativity. These models excel in creating digital artwork that rivals the finesse of human artists, marking a significant advance in AI capabilities. While these models' proficiency in crafting digital art has been acknowledged, their potential to address critical challenges in machine learning, specifically data scarcity, is only beginning to be explored. This project is an investigation into the potential of these generative diffusion models to augment training data, thereby enhancing the generalizability of sophisticated machine-learning models.

Data augmentation is a common strategy in numerous deep-learning methodologies, offering a solution when faced with limited training data. This process involves increasing the quantity of training examples by performing operations like rotation, reflection, cropping, translation, and scaling on existing images, potentially enriching the dataset exponentially. By adding synthetic yet realistic images to the training data, the likelihood of overfitting is significantly reduced. This process not only increases the accuracy but also bolsters the generalization ability of deep learning models. This becomes particularly vital in the context of convolutional neural networks (CNNs), which struggle with learning rotationally invariant features unless provided with sufficient examples at varied rotations in the training data. Given these considerations, the project explores the potential of using a diffusion model, specifically OpenAI's DALL-E, to model the underlying distribution of training data. This method enables the generation of additional synthetic data, which can be used to augment the real training data, presenting an innovative approach to the issue of data scarcity.

The central issue that this project addresses is the limitation of available data for training intricate convolutional neural networks (CNNs). In image classification tasks, the availability of large and diverse datasets is crucial for developing models that can effectively generalize to unseen data. However, assembling such large and varied datasets is often challenging and resource-intensive. This project utilizes a constrained dataset of cat and dog images, proposing to augment this limited dataset through the use of generative AI and style transfer techniques.

The project employs machine learning algorithms in a two-pronged strategy. Initially, a CNN is used to establish a benchmark model using the original, limited dataset. Subsequently, the capabilities of OpenAI's DALL-E, a generative diffusion model, are harnessed to create three distinct variations of each image in the dataset. These generated images are then incorporated into the training data for a second CNN. Following this, style transfer techniques are applied to the original and generated images, generating a diverse dataset to train a third CNN. This process thus explores the synergy of generative AI and style transfer in augmenting training data.

II. Related Work

In the field of machine learning, particularly in image classification, many researchers are striving to address the persistent challenge of model robustness when faced with naturally shifted data distributions. There's a common struggle with classifiers' performance degrading when trained on datasets similar to the test set, rather than those from varied distributions like sketches or animations.

The paper "Leaving Reality to Imagination: Robust Classification via Generated Datasets" [1] presents a novel solution to improving the robustness of neural image classifiers by using generated datasets. This method utilizes generated datasets to bridge the commonly observed performance gap when classifiers are trained on datasets that mirror the test set closely.

The authors propose using generated datasets created by a generative model known as the Stable Diffusion (SD) model. The SD model is a text-to-image generative model that has been trained on a large, diverse dataset named LAION, and it is

capable of generating novel images based on natural language descriptions. This capability allows the model to create new images, realistically combine unrelated concepts, and apply novel transformations to existing images. The authors propose three generation strategies using the SD model: generation with class labels, generation with source images, and generation with both labels and images.

These strategies create synthetic data that is more varied and representative than those produced by conventional augmentation methods. The authors demonstrate that classifiers trained on a combination of real data and this generated data perform better, in terms of accuracy and robustness, than those trained with standard methods or popular augmentation strategies, even when subjected to natural distribution shifts. This result indicates that generated datasets offer a promising solution for improving the robustness of neural image classifiers. However, this strategy is computationally demanding due to the complexity of the generative models used and the large volume of data generated. Future research should focus on streamlining this process and determining the optimal balance between real and generated data for training robust classifiers.

In another paper titled ”GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks” [2] addresses the challenge of data scarcity in medical imaging machine learning applications by using Generative Adversarial Networks (GANs) to generate synthetic data. The lack of large, labeled datasets, due to the costly and time-consuming nature of medical image annotation, can inhibit the performance of supervised machine learning algorithms. The introduction of GAN-derived synthetic data to the training datasets used in brain segmentation tasks led to improvements in the Dice Similarity Coefficient.

The study used the Progressive Growing of GANs (PGGAN) architecture for its training stability at large image sizes and robustness to hyperparameter selection. This GAN was trained on 80,000 patches from the available training dataset before training a segmentation Convolutional Neural Network (CNN). An alteration to the default PGGAN architecture, the addition of a 32x32 layer of Gaussian noise, resulted in more realistic CT images.

Experiments assessing the effect of introducing GAN-derived synthetic data to a segmentation task modified several key variables, including the amount of available real data. This approach shows promise for augmenting training datasets in medical imaging, but the choice of GAN architecture may impact the quality of the augmentation.

III. Our Solution

This project draws upon three distinct datasets, all of which focus on images of cats and dogs. The exploration of these datasets involves several stages of data preprocessing and feature engineering, aiming to optimize the input for our convolutional neural network models.

A. Description of the Dataset

The initial dataset comprises a total of 600 images, divided equally between photos of cats and dogs. This dataset serves as the foundation for the project, establishing the baseline upon which we built and evaluated our subsequent data augmentation strategies. The images in this dataset were sourced from an online collection of pet images, curated to ensure a balance of breeds and poses for both the cat and dog categories.

Fig. 1: Original Dataset Samples

To prepare the images for the CNN model, they were first resized to a uniform dimension of 128x128 pixels, as required by the model. The pixel intensities were also rescaled to a range of 0 to 1 from the standard 0-255 range. To further enhance the training data, the images were subjected to various transformations such as shear, zoom, and horizontal flip. These transformations diversified the data, providing variations in perspective and orientation, and helped to prevent overfitting during training.

No significant outliers or missing features were detected in this initial dataset. The images were also batched into sets of 32, and their labels were binary-encoded (0 for cats, 1 for dogs) for the training process.

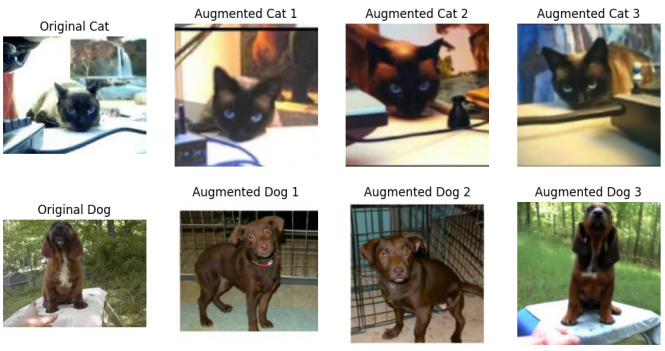

Fig. 2: Dalle Augmented Dataset Samples

Our second dataset expands upon the first, using OpenAI’s DALL-E to generate three unique variations of each image in the initial dataset. This process resulted in an augmented dataset of 1800 images, providing a more comprehensive array of visual data for the training of our second CNN. These DALL-E generated images maintained the original labels of their source images, ensuring continuity in the data representation. An important aspect of data preprocessing at this stage

involved ensuring the generated images' quality and relevance. Any images that failed to accurately represent a cat or a dog were removed from the dataset.

Fig. 3: Style Transfer Augmentation Example

The third dataset represents the culmination of our data augmentation process. It incorporates all images from the original and DALL-E augmented datasets, with an additional set of style-transferred versions of each image. The style transfer technique was applied using a pre-trained neural style transfer model from TensorFlow, creating a visually diverse dataset that further enhances the complexity and variability of the training data. The final dataset, therefore, comprises a total of 5400 images, providing a rich and varied source of data for the training of our third CNN. Any images that failed to accurately represent a cat or a dog were removed from the dataset.

B. Implementation Details

The project involves the development and comparison of three distinct CNN models, each trained on a different dataset. The three datasets are the original image dataset, the DALL-E augmented dataset, and the dataset resulting from the application of neural style transfer techniques. This section elaborates on the implementation details, performance testing, hyperparameter tuning, and model selection process for these CNNs. The first CNN was designed and trained on the initial dataset comprising 600 images of cats and dogs. The model was developed using the Keras library, a popular high-level neural networks API. The architecture of the CNN consists of several layers, each serving a specific function.

The first layer of the model is a 2D convolutional layer, equipped with 32 filters of 3x3 size and the ReLU (Rectified Linear Unit) activation function. The input to this layer is an image of size 128x128 pixels with three color channels. This layer is followed by a 2x2 max pooling layer that reduces the spatial dimensions of the output from the preceding convolution layer, thereby retaining only the most important information. A dropout layer with a dropout rate of 0.25 follows next. This layer randomly nullifies 25% of its input units during training to prevent overfitting. This combination of a convolution layer, max pooling layer, and dropout layer is repeated twice more, each time increasing the number of filters in the convolution layer (64 and 128, respectively) to allow the network to learn more complex patterns. After these layers, the output from the last max pooling layer is flattened into a one-dimensional vector to prepare it for the fully connected layers. The flattened output is fed into a Dense layer with 128 neurons, which uses the ReLU activation function. A dropout layer with a dropout rate of 0.5 follows this, further aiding in the reduction of overfitting. The final layer is another Dense layer, this time with a single neuron, and it uses a sigmoid activation function. This final layer outputs the predicted probability for the binary classification task.

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d_8 (MaxPooling 2D)	(None, 63, 63, 32)	0
dropout (Dropout)	(None, 63, 63, 32)	0
conv2d_9 (Conv2D)	(None, 61, 61, 64)	18496
max_pooling2d_9 (MaxPooling 2D)	(None, 30, 30, 64)	0
dropout_1 (Dropout)	(None, 30, 30, 64)	0
conv2d_10 (Conv2D)	(None, 28, 28, 128)	73856
max_pooling2d_10 (MaxPoolin g2D)	(None, 14, 14, 128)	0
dropout_2 (Dropout)	(None, 14, 14, 128)	0

...
Total params: 3,304,769
Trainable params: 3,304,769
Non-trainable params: 0

Fig. 4: CNN Architecture

After the architecture is set up, the model is compiled using the Adam optimizer and the binary cross-entropy loss function. Accuracy is the metric used to evaluate the model's performance. Data augmentation techniques, such as pixel normalization, shear transformations, zooming, and horizontal flips, are applied to the training data using Keras's ImageDataGenerator. This technique helps prevent overfitting and enables the model to generalize better. Finally, the model is trained for 50 epochs using the training data, with the performance validated using a test set. The training is configured such that each epoch consists of 18 steps, and validation is performed over 2000 steps.

IV. Results

The performance evaluation of the three implemented Convolutional Neural Networks (CNNs) provides intriguing insights. Each CNN was trained on distinct datasets with varying degrees of augmentation: the original dataset, the original dataset combined with DALL-E generated images, and a comprehensive set combining original, DALL-E augmented, and style transfer variations of the images. It's important to note that the test set, against which these models were evaluated, consisted of real images of cats and dogs that were included in the original dataset but not used in training.

The first CNN, trained solely on the original dataset, achieved an accuracy of 0.877%. This baseline performance is commendable given the limited size of the training dataset this accuracy score will be the baseline for comparing if the augmented datasets aid in the robustness of the CNN.

The second CNN, which incorporated DALL-E generated images into the training set, recorded an accuracy of 0.842%. Interestingly, this accuracy is slightly lower than that of the first CNN. A plausible explanation for this could be that the DALL-E generated images, although they increase the overall size of the dataset and introduce variations, might contain features or styles that aren't well-represented in the original, real-world dataset. This potentially makes the model less effective at generalizing to unseen real-world data in the test set. Another possible explanation for the reduced accuracy could be that the DALL-E generated images were too similar to the original training data which led to overfitting.

The third CNN, trained on a comprehensive dataset that incorporated style transfer augmentations, demonstrated a marked improvement, achieving an accuracy score of 0.9086%. The inclusion of style transfer augmentations appears to have enhanced the model's ability to generalize, likely by introducing a broader range of visual features for the model to learn.

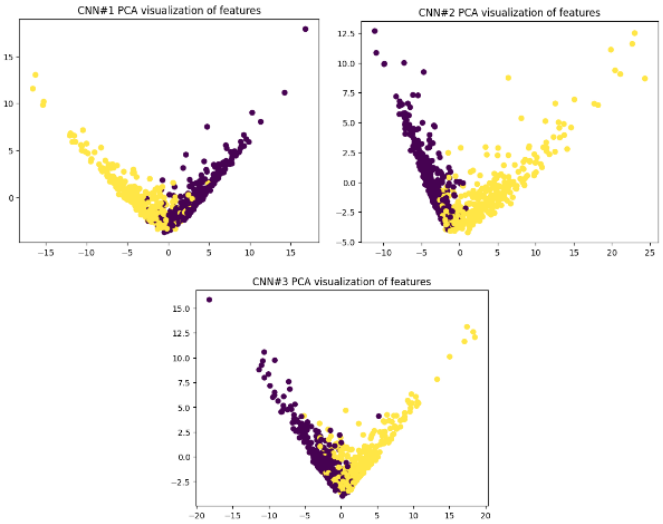

Fig. 5: PCA Visualization of Features

PCA visualizations of the features learned by each model provide further insights. By comparing these visualizations, we can better understand how the different data augmentation techniques influenced the models' learning processes and their ability to extract meaningful features from the data.

Maximum Mean Discrepancy (MMD) is a statistical technique used to compare distributions of activations between two models. In this study, MMD was employed to assess the similarity of the feature representations learned by the three implemented CNNs. The MMD results showed that the activations of the three CNNs were not identical, but their differences were relatively small. The MMD score between the first and second CNNs was 0.00148, while the score between the first and third CNNs was 0.00215, and the score between the second and third CNNs was 0.00235. These results suggest that the different data augmentation techniques applied to the training sets had an impact on the feature representations learned by the CNNs. This finding is consistent with the PCA visualizations, which also showed that the feature representations of the three CNNs were similar.

Overall, these results highlight the potential benefits of using advanced generative AI techniques, such as DALL-E and neural style transfer, for data augmentation, especially in scenarios where the original dataset may be limited. However, they also underscore the need for careful selection and evaluation of synthetic data added to the training set, as it can significantly influence the model's ability to generalize to new, real-world data.

V. Future Directions

Looking ahead, there are several promising directions for further improving the performance of our approach. Firstly, a potential avenue for exploration involves the incorporation of more advanced diffusion models, such as Midjourney. Unlike the current neural style transfer technique used in our project, Midjourney allows for the integration of reference images which can then be modified with accompanying text prompts. This could provide a more controlled, context-aware method for data augmentation, generating synthetic images that better represent the diversity and intricacy of the real-world data, and thus improving the generalizability of the trained models.

Secondly, another enhancement to our approach could be the inclusion of a Convolutional Variational Autoencoder (CVAE) in our pipeline. A CVAE is a type of generative model that learns a compact, dense representation of the input data in a latent space, and uses this learned representation to generate new data. The convolutional nature of the CVAE makes it particularly well-suited to image data. By training a CVAE on our datasets, we could generate additional synthetic images that retain the core visual characteristics of the original data but also introduce novel variations. This could particularly aid in handling edge cases, where the CNN might struggle due to the lack of sufficient representative data in the training set.

Finally, we could also explore different architectures and configurations for the CNN, including alternative activation functions, regularizations, or even more advanced models such as ResNets or DenseNets. Furthermore, experimenting with different loss functions and optimization algorithms might yield improvements in model performance. With these prospective improvements, we are confident that our approach can be further refined, pushing the boundaries of what generative AI can achieve in the realm of data augmentation for image classification tasks.

VI. Conclusion

This project demonstrated that integrating generative AI techniques into the data augmentation process improved the performance of convolutional neural networks in image classification tasks. The most effective model was the third CNN, trained on a dataset augmented with synthetic images generated by DALL-E and enhanced with neural style transfer techniques. It achieved an accuracy of 90.86% on the test set, surpassing the other models.

Our findings suggest that using generative AI models for data augmentation can be a beneficial strategy, particularly

in contexts with limited data. However, the most suitable algorithm largely depends on the specifics of each task. Future work could investigate more advanced diffusion models and generative models to further enhance performance. The synergy between generative AI and convolutional neural networks unveiled in this project demonstrates a promising avenue for model improvement and generalizability.

VII. Running the code

To replicate the experiments conducted in this research, the code is available at the following GitHub repository: https://github.com/kyritzb/Synthetic-Data-Classification-Experiment. The repository contains three main Jupyter Notebook files: data.ipynb, styletransfer.ipynb, and main.ipynb, each serving a specific purpose in the experimentation process.

data.ipynb: This notebook file allows the generation of variations of the original images using OpenAI's DALL-E model. By executing the code in this notebook, you will be able to augment the dataset with synthetic images that mimic the creativity of human artists. These generated images serve as an additional source of training data for the subsequent convolutional neural network models.

styletransfer.ipynb: In this notebook, the images from the original dataset, as well as the DALL-E augmented dataset, can be transformed into new styles using neural style transfer techniques. By running the code provided in this notebook, you can convert the images into diverse artistic styles, further enriching the dataset's complexity and variability. The resulting style-transferred images are instrumental in training the third convolutional neural network.

main.ipynb: The primary notebook file, main.ipynb, focuses on the training and evaluation of the convolutional neural network models. By executing the code in this notebook, you can run the CNN training process using the augmented datasets created through DALL-E and style transfer techniques. The notebook allows for a comprehensive comparison of the performance of the three CNN models trained on different datasets, highlighting the impact of generative AI on image classification tasks.

To reproduce the experimental results, it is crucial to ensure that the necessary dataset is available. The README file in the GitHub repository provides instructions on how to download the original dataset of cat and dog images used in this research. Please follow the guidelines provided in the repository's README to acquire the dataset and ensure its proper placement within the project directory structure.

By following the steps outlined in the Jupyter Notebook files and ensuring the availability of the required dataset, researchers can successfully run the code and replicate the experiments conducted in this study.

References

[1] H. Bansal and A. Grover, "Leaving reality to imagination: Robust classification via generated datasets," 2023.

[2] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw, and D. Rueckert, "Gan augmentation: Augmenting training data using generative adversarial networks," 2018.