ENHANCING IMAGE DATA WITH DIFFUSION-BASED AUGMENTATION TECHNIQUES

Savya Khosla
savyak2

Shrey Sarswat
sarswat2

Rishabh Garg
rg18

1 Cover Page

The problem addressed in this project is how to generate more diverse and realistic data to improve the performance of deep learning models. We investigate the use of diffusion models, a recently proposed generative model, for data augmentation and compare their performance with other commonly used techniques such as RandAugment, AutoAugment, MixUp, CutOut, and CutMix.

Our line of attack involves using an off-the-shelf diffusion model to generate training dataset, which is then used to train a ResNet-based model. The system architecture for augmenting data involves adapting the diffusion model to images of new concepts through multi-class textual inversion. The system generates M augmented versions of each image in the dataset using a pretrained stable diffusion checkpoint.

We find that diffusion-based data augmentation coupled with MixUp can outperform other techniques in terms of accuracy and generalization on the chosen dataset, especially when the amount of training data is limited. The results show an improvement of more than 5% over the baseline on the PASCAL VOC dataset. These results suggest that diffusion models have the potential to become a valuable tool for data augmentation in deep learning.

A set of augmented examples generated using the diffusion model can be seen at the following link: Google Drive

Our codebase is at the following link: Google Drive

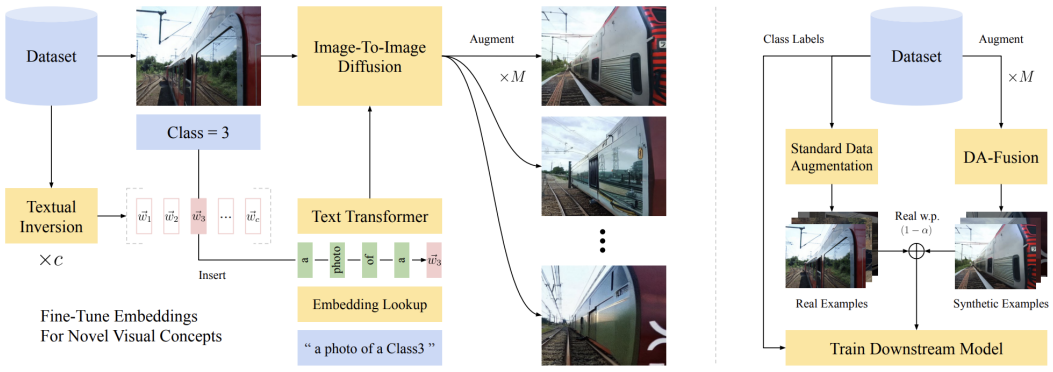

Figure 1: The system architecture involves the use of a diffusion model that is adapted to images of new concepts through multi-class Textual Inversion. The system generates M augmented versions of each image in the dataset using a pretrained Stable Diffusion checkpoint. During the training of downstream models, the synthetic images are mixed with real data.

2 Introduction

One of the most critical challenge in deep learning is the lack of data for domain-specific applications. Collecting and annotating a large amount of high-quality data is a time-consuming and expensive process, especially in domains where manual labeling is required, such as medical imaging. Data augmentation is one of the most well adopted techniques used to deal with such problem. However, conventional data augmentation techniques like rotation and transformation, often result in unrealistic and repetitive data.

On the other hand, diffusion models [1] are a type of generative model that have become increasingly popular in recent years due to their ability to generate high-quality, diverse, and realistic data. Unlike other generative models, such as GANs and VAEs, diffusion models work by iteratively updating a noise tensor to generate the final output using a sequence of learned diffusion steps. They have shown promise in a variety of applications, including image and video generation, data augmentation for deep learning, and density estimation. Diffusion models are also known for their ability to produce sharp and high-resolution images, making them particularly useful for tasks that require generating realistic and detailed images.

In this project, we reproduce the model given in the *Effective Data Augmentation With Diffusion Models* [2]. To further understand the usability of diffusion models in such scenarios, we analyze and compare how well a baseline model performs on data augmented via diffusion with respect to data augmented via other conventional methods in literature. In particular, we use RandAugment [3], AutoAugment [4], CutOut [5], CutMix [6], and MixUp [7] as our baseline.

2.1 Background

Diffusion models are a type of sequential latent variable models that generate data samples via a Markov chain with learned Gaussian transitions, called the reverse process, starting from an initial noise distribution. The transitions are designed to gradually reduce noise according to a schedule $\beta_1, ..., \beta_T$ such that the final sample represents a sample from the true data distribution. Transitions are often parameterized by a fixed covariance $\Sigma_t = \beta_t I$ and a learned mean $\mu_\theta(x_t, t)$ that is defined as a function of the current sample x_t and the current time step t. This parameterization choice results from deriving the optimal reverse process, where a neural network parameterized by weights θ is trained to process a noisy sample x_t to predict the noise added to real images by the forward process.

Given real samples x_0 and noise $\epsilon \sim \mathcal{N}(0, I)$, the forward process can be sampled at an arbitrary timestep using equation 1.

$$x_t = \sqrt{\tilde{\alpha_t}} x_0 + \sqrt{1 - \tilde{\alpha_t}} \epsilon \tag{1}$$

Here, $\alpha_t = 1 - \beta_t$ and $\tilde{\alpha_t} = \prod_{s=1}^{t} \alpha_t$.

Following [2], we use a pre-trained stable diffusion model [8], which includes a text encoder that enables text-to-image generation.

2.2 Related Work

Data Augmentation: To prevent overfitting, deep neural networks usually require a substantial quantity of training data, which can be achieved by data augmentation. One popular approach is to apply geometric transformations to the input data, such as rotation, scaling, and flipping, which can help the model learn to recognize objects from different viewpoints and orientations. In [9], random cropping and horizontal flipping are used to augment the image data, leading to a significant improvement in the classification accuracy.

Another approach is to apply various noise and distortion operations to the input data, such as adding Gaussian noise, blurring, or occlusion, which can help the model learn to be more robust to noise and variations in the input. In [10], the authors propose a method called label smoothing and mixup, which combines different training examples with random weights, to augment the training data and regularize the model.

More recently, generative models, such as generative adversarial networks (GANs), variational autoencoders (VAEs) and diffusion models, have been used for data augmentation by generating realistic synthetic examples that can increase the diversity and quantity of the training data. In [11], the authors propose a method called "Wasserstein augmentation" which uses a GAN to generate additional training examples that can improve the performance of the model.

Diffusion Models: In recent years, diffusion models have gained increasing attention for their outstanding performance in various applications, including image generation, video production, and molecule design. In this project, we

explore the capabilities of diffusion models and their potential for diverse applications. Our study was informed by previous research, including the works of [12] and [13], which provided a comprehensive overview of the methods and applications of diffusion models. Building on these insights, we plan to utilize a pre-trained diffusion model to synthesize augmented views of images, as demonstrated in [2]. This approach addresses the issue of limited diversity in traditional data augmentation techniques, such as rotation, cropping, and scaling, by leveraging a text-to-image diffusion model to parameterize image-to-image transformations.

Image Editing: Several approaches in literature discuss image editing using diffusion models. These include inpainting with diffusion [12, 14], modifying attention weights of the diffusion process [15, 16], and SDEdit [17] which inserts real images partway through the reverse diffusion process. While SDEdit has been used for generating synthetic data, the proposed method differs from DA-Fusion in that it adapts the generative model to new visual concepts and considers augmentations to individual objects.

2.3 Dataset

The PASCAL Visual Object Classes (VOC) dataset [18] is a widely used benchmark dataset for object detection, segmentation, and classification tasks in computer vision research. The dataset consists of 20 object categories, such as person, car, and cat, and contains over 11,000 images with annotated object bounding boxes and pixel-level segmentation masks. The annotations are based on the consensus of multiple human annotators, ensuring high quality and accuracy. We will use this data for 20-way object classification task.

3 Methods

In this study, we investigate the efficacy of various data augmentation techniques on the PASCAL VOC classification task. Specifically, we aim to analyze the performance of the following augmentation methods:

3.1 RandAugment [3]

RandAugment is a popular and effective data augmentation technique for image classification tasks. It applies random and sequential image transformations to generate diverse and augmented versions of the original images, improving model accuracy and robustness.

Figure 2: The base image (leftmost) and their different augmented versions using RandAugment.

3.2 AutoAugment [4]

AutoAugment is a data augmentation technique that uses reinforcement learning to automatically find the optimal set of image transformations for a given task. While it has achieved great performance on various image classification tasks, this approach is often not suitable for some practical applications due to its computational complexity. For our experiments, we use the Imagenet policy for applying augmentations.

Figure 3: The base image (leftmost) and their different augmented versions using AutoAugment.

3.3 MixUp [7]

Mixup is a data augmentation technique that combines pairs of training samples to create new augmented samples, encouraging the model to learn more generalizable decision boundaries. It has been shown to improve model accuracy and is computationally efficient, making it a popular choice for practical applications. It's implemented with the following formulas:

$$x = \lambda x_i + (1 - \lambda) x_j \tag{2}$$

$$y = \lambda y_i + (1 - \lambda) y_j \tag{3}$$

where x_i and x_j are the inputs and y_i and y_j are their one-hot encodings.

Figure 4: An image-label pair derived from applying the MixUp transformation to 'person' and 'aeroplane' class

3.4 CutOut [5]

Cutout involves removing a random patch of pixels from a training image to create a new augmented image. The size and position of the patch are chosen randomly based on a predetermined distribution. The label for the new augmented image remains the same as the original image.

Figure 5: Images obtained after applying CutOut transformation.

3.5 CutMix [6]

CutMix works by selecting two random images from the training set, cutting a randomly sized patch from one image, and pasting it onto another image. The size of the patch and the position of the cut are chosen randomly based on a predetermined distribution. The label for the new augmented image is determined by taking a weighted average of the labels of the two original images based on the area of overlap between the pasted patch and the original image.

$$x = \mathbf{M} \odot x_i + (\mathbf{1} - \mathbf{M}) \odot x_j \tag{4}$$

$$y = \lambda y_i + (1 - \lambda) y_j \tag{5}$$

where $\mathbf{M} \in \{0, 1\}^{W \times H}$, $\odot$ is element-wise multiplication, x_i and x_j are the inputs, and y_i and y_j are their one-hot encodings.

Figure 6: An image-label pair derived from applying the CutMix transformation to 'person' and 'sheep' class of PASCAL dataset.

3.6 DA-Fusion [2]

DA-Fusion is a diffusion-based image augmentation technique that

1. applies to all types of images
2. systematically balances real and sythetic data
3. allows stacking of different augmentation techniques
4. provides control on what parts of images should be augmented

Modelling Novel Views

DA-Fusion works by inserting new tokens into a generative model's vocabulary and fine-tuning their embeddings using the standard diffusion loss as given by 6. This approach allows for the generation of plausible augmentations for images with elements that the diffusion model cannot generate out-of-the-box. Instead of generating synthetic images from scratch, DA-Fusion uses image-to-image transformations to splice real images into the reverse diffusion process. This approach enables the generation of plausible augmentations even for novel concepts with limited labelled examples.

$$L_D = \mathbb{E}[||\epsilon - \epsilon_\theta(\sqrt{\tilde{\alpha_t}}x_0 + \sqrt{1-\tilde{\alpha_t}}\epsilon, t)||^2] \tag{6}$$

Balancing Real and Synthetic Data

One caveat of using diffusion models for data augmentation is that training models on synthetic datasets generated from generative models can lead to overemphasizing spurious qualities in the data. So, to balance real and synthetic data, [2] adopts a method that assigns different sampling probabilities to real and synthetic images. They generate M augmentations for each real image, resulting in a synthetic dataset. The probability α determines whether a real or synthetic image is added to the minibatch. [2] states that $\alpha = 0.5$ works effectively in all domains.

Stacking Augmentations

To maximize the diversity of augmented images, DA-Fusion uses a sequence of stackable data augmentations based on generative models. The process works by defining a sequence of tuples that consist of image transformations and activation probabilities, which can be randomly sampled to generate diverse augmentations. To ensure diversity, the image transformations should operate with different granularity on the image. To this end, DA-Fusion turns the existing methodology of SDEdit [17] into a stackable data augmentation method.

Object-Centric Augmentations

Lastly, DA-Fusion also applies transformations at the object level by leveraging in-painting to insert content into the diffusion process at locations specified by a pixel-wise mask as shown in equation 7. This allows for independent transformations that separate objects and backgrounds, and is more effective than traditional data augmentations which violate this principle. The proposed method can be applied to all images regardless of class and content and can be stacked with other augmentations. Figure 7 (taken from [2]) demonstrates such object-centric augmentation.

$$x_t \leftarrow (1-v) \odot x_t + v \odot (\sqrt{\tilde{\alpha_t}}x_0^{\text{ref}} + \sqrt{1-\tilde{\alpha_t}}\mu) \tag{7}$$

Here, x_0^{ref} is the source image to inpaint, $v \in [0,1]^{W \times H}$ is the pixel-wise mask specifying which content of the source is to be modified, $\mu \sim \mathcal{N}(0, I)$, and $\odot$ represents element-wise multiplication.

Figure 7: Object-centric augmentations that can be applied independently to each segment of the image.

4 Results

We evaluate the effectiveness of 6 data augmentation techniques, including RandAugment, AutoAugment, MixUp, CutOut, CutMix, and DA-Fusion. Further, we experiments with different parameters of DA-Fusion and also combine it with the second best augmentation technique.

4.1 Experimental Setup

For all our experiments we use the PASCAL VOC dataset and ResNet18 backbone. The ResNet18 model is initialized with the imagenet weights and its last layer is modified to have 20 instead of 1000 units. We use a learning rate of 1×10^{-4} for the ResNet backbone and 5×10^{-3} for the final FC layer. To ensure a fair comparison, we use the same hyperparameters while experimenting with different augmentation techniques. In particular, we use a batch size of 128, SGD optimizer with a momentum of 0.9, cosine annealing learning rate schedule, and train the model for 35 epochs. All our experiments are done on Google Colab and we use PyTorch framework to code our training, testing and data loading pipelines.

Hyperparameters for the specific augmentations are as follows:

Baseline: The baseline represents a ResNet18 model trained on unaugmented PASCAL VOC images.

RandAugment: We use PyTorch implementation of the RandAugment with num_ops $= 2$ and magnitude $= 1$.

AutoAugment: We use PyTorch implementation and set the policy to AutoAugmentPolicy.IMAGENET.

MixUp: For MixUp, we follow the code given at [19]. We use $\alpha = 0.5$, where α is the parameter of the Beta distribution used for MixUp.

CutOut: We follow the code given at [20] and use n_holes $= 1$ and length $= [50, 150]$, where n_holes is the number of patches to cut out of each image and length is length (in pixels) of each square patch.

CutMix: We follow the code given at [19] and use $\alpha = 0.5$, where α is the parameter of the Beta distribution used for CutMix.

DA-Fusion: Table 1 summarizes the hyperparameters used for DA-Fusion. We use the same hyperparameter values as used by the authors of [2].

Table 1: Hyperparameter values for DA-Fusion

Hyperparameter	Value
Synthetic Probability α	0.5
Stacked Augmentations k	4
Activation Probabilities p_i	$1/k$
Synthetic Images Per Real M	10
Textual Inversion Token Initialization	"the"
Textual Inversion Batch Size	4
Textual Inversion Learning Rate	0.0005
Textual Inversion Training Steps	1000
Class Agnostic Prompt	"a photo"
Textual Inversion Prompt	"a photo of a ClassX"
Real Guidance Strength t_0	0.5
Stable Diffusion Checkpoint	CompVis/stable-diffusion-v1-4
Stable Diffusion Guidance Scale	7.5
Stable Diffusion Resolution	512
Stable Diffusion Denoising Steps	1000
Classifier Architecture	ResNet18
Classifier Learning Rate	0.0001
Classifier Batch Size	32
Classifier Training Steps	10000
Classifier Early Stopping Interval	200

4.2 Results

Using the experimental setup defined above, we train the ResNet18 model using all the augmentation techniques and the results are shown in Table 2. For DA-Fusion, we generate 10 augmented views for γ ($= 4$ or 8) examples per class. That is, we randomly select γ images from each class and generate 10 augmentations for each image.

Table 2: Result of different augmentation techniques on the classification performance

Augmentation Technique	Test accuracy
Baseline	72.21%
RandAugment	73.50%
AutoAugment	71.87%
MixUp	77.47%
CutOut	72.92%
CutMix	76.66%
DA-Fusion (γ = 4)	73.55%
DA-Fusion (γ = 8)	73.64%
DA-Fusion + MixUp	**78.35**%

5 Discussion and Conclusions:

In this study, we have explored the use of diffusion models for data augmentation in deep learning. This approach addresses the problem of limited diversity in classical data augmentation techniques (such as rotation, cropping, and scaling) by utilizing pre-trained text-to-image diffusion models to parameterize image-to-image transformations. In particular, we reproduced the model described in [2] and compared its performance with conventional data augmentation methods. Our results show that the diffusion-based data augmentation method outperforms the conventional methods in terms of accuracy, diversity, and realism of the generated images.

Our results suggest that diffusion-based data augmentation can be an effective way to improve the performance of deep learning models. The advantage of using diffusion models for data augmentation is their ability to generate high-quality and diverse data that better mimics the real-world distribution. It can help overcome the limitations of conventional data augmentation techniques and enable us to train models with limited data. We believe that this study provides valuable insights into the potential of diffusion models in data augmentation.

6 Statement of Individual Contribution

All team members contributed equally to the report and hyperparameter tuning. In particular,

Savya Khosla (savyak2) worked on the following:

1. setting up the baseline
2. generating results for MixUp and CutMix
3. generating results for DA-Fusion+MixUp experiment

Shrey Sarswat (sarswat2) worked on the following:

1. generating results for AutoAugment and CutOut
2. setting up the DA-Fusion pipeline
3. generating results for DA-Fusion ($\gamma = 4$)

Rishabh Garg (rg18) worked on the following:

1. setting up the data loader pipeline
2. generating results for RandAugment
3. generating results for DA-Fusion ($\gamma = 8$)

References

[1] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

[2] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models, 2023.

[3] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

[4] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data, 2019.

[5] Terrance DeVries and Graham Taylor. Improved regularization of convolutional neural networks with cutout. 08 2017.

[6] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019.

[7] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[10] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[11] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

[12] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2021.

[13] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2022.

[14] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022.

[15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022.

[16] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models, 2022.

[17] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations, 2022.

[18] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. 88(2):303–338, jun 2010.

[19] pytorch/vision. https://github.com/pytorch/vision/blob/main/references/classification/transforms.py.

[20] raeidsaqur TDeVries. Cutout. https://github.com/uoguelph-mlrg/Cutout.