Mind the (domain) gap: Metrics for the differences in synthetic and real training data distributions

Ashley S. Dale[a], William R. Reindl[b], Edwin Sanchez[b], Albert William[c], and Lauren Christopher[a]

[a]Purdue School of Engineering and Technology, Address, Indianapolis, U.S.A.
[b]IUPUI School of Science, Address, Indianapolis, U.S.A.
[c]Luddy School of Informatics, Computing, and Engineering, Address, Indianapolis, U.S.A.

ABSTRACT

Synthetic data are frequently used to supplement a small set of real images and create a dataset with diverse features, but this may not improve the equivariance of a computer vision model. Our work answers the following questions: First, what metrics are useful for measuring a domain gap between real and synthetic data distributions? Second, is there an effective method for bridging an observed domain gap? We explore these questions by presenting a pathological case where the inclusion of synthetic data did not improve model performance, then presenting measurements of the difference between the real and synthetic distributions in the image space, latent space, and model prediction space. We find that augmenting the dataset with pixel-level augmentation effectively reduced the observed domain gap, and improves the model F1 score to 0.95 compared to 0.43 for un-augmented data. We also observe that an increase in the average cross entropy of the latent space feature vectors is positively correlated with increased model equivariance and the closing of the domain gap. The results are explained using a framework of model regularization effects.

Keywords: synthetic data generation, domain gap, latent space features, data augmentation, latent feature representation, synthetic training data

1. INTRODUCTION

Synthetic data are frequently used to augment a dataset with the goal of improving a model's ability to generalize to unseen data.[1] This generalization is commonly referred to as *invariance* and *equivariance*; *equivariance* refers to a change model's output correctly correlated to a change of the model's input, and *invariance* is a special case of equivariance referring to no change in the model's output given a task-irrelevant change to the model's input.[2–4] In this work, recent efforts[2] to establish a theoretical understanding of invariance and equivariance are used to motivate a practical analysis of the model's **training data**, with the hypothesis that a data augmentation method that causes a model's generalization performance to decrease can be attributed to a domain gap in the training data distribution.[5,6]

Our measurement of such a domain gap has the form of a pairwise similarity comparison

$$\mathcal{E}(\mathcal{X}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j>i}^{n} m(x_i, x_j) \tag{1}$$

where x is an instance of the dataset X (with n instances) possessing a domain gap, and $m(x_i, x_j)$ is a similarity measure between the i^{th} and j^{th} instances. The domain gap can be quantified in the input image domain, the feature space domain, and the prediction domain as shown in Fig. 1; what then is the best choice of similarity measure m?

Further author information: (Send correspondence to L.C.)
L.C.: E-mail: lauchris@iu.edu
A.S.D.: E-mail: asdale@purdue.edu

Accordingly, this work explores the following questions: **First**, what measures may be useful for measuring a domain gap between real and synthetic training data distributions? **Second**, is there an effective method for bridging an observed domain gap? They are explored by analyzing a previously reported[5] pathological case where augmenting a small real world (RW) image dataset with synthetic image data manufactured from a virtual world (VW) did not improve model performance; the F1 score of a transfer-learning trained YOLOv8[7] model dropped 16% from 0.51 to 0.43. Five different state-of-the-art mixed sample data augmentation (MSDA) techniques were tested to bridge the domain gap between the RW and VW data, including variations of CutMix,[8] CutOut,[9] Mixup,[10] pure noise,[11] and our own contribution Mixed Feature Data Augmentation (MFDA) based on Manifold Mixup.[12] We then quantify the effectiveness of the augmentation technique at bridging the domain gap by evaluating the average similarity of the newly augmented dataset according to Eq. 1, where different similarity measures m are evaluated according to the criteria discussed in §4.4.

The following section, §2, presents the main results of the paper, followed by a discussion in §3 and a summary in §5. The methods are presented in §4. The results presented in this paper are not exhaustive. Rather, they are designed to provide intuition regarding which similarity measures may provide insight into the source and correction of domain gaps.

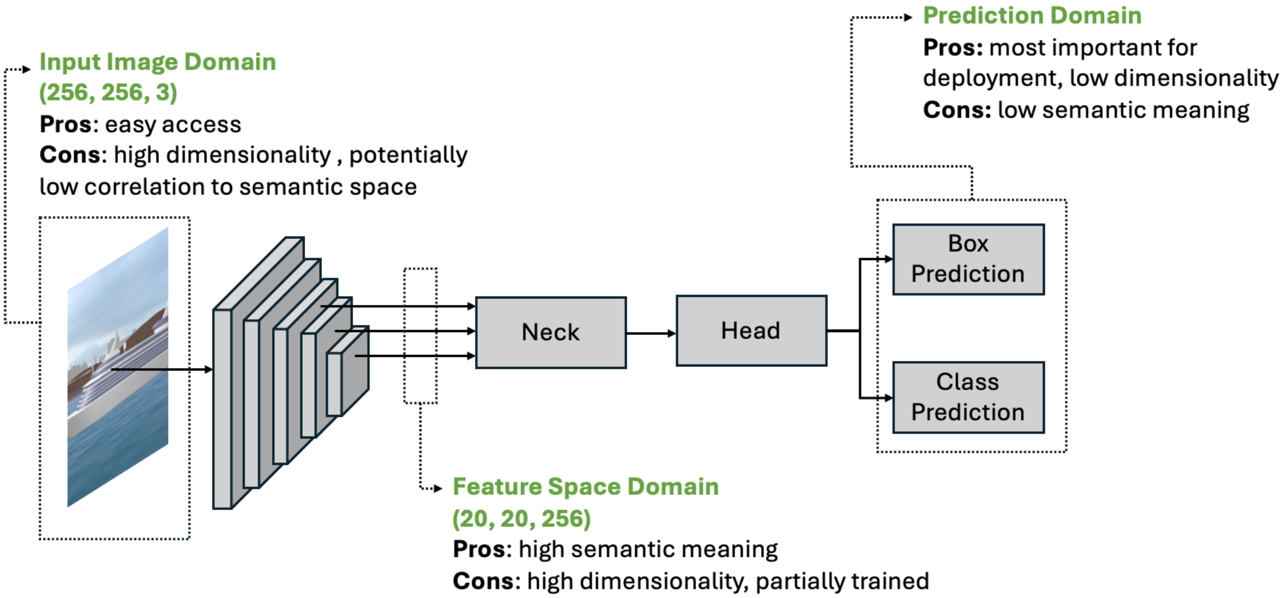
Input Image Domain
(256, 256, 3)
Pros: easy access
Cons: high dimensionality , potentially low correlation to semantic space
Prediction Domain
Pros: most important for deployment, low dimensionality
Cons: low semantic meaning
Neck
Head
Box Prediction
Class Prediction
Feature Space Domain
(20, 20, 256)
Pros: high semantic meaning
Cons: high dimensionality, partially trained

Figure 1. Possible locations to measure domain gaps in a YOLOv8[7] style model: input image space (raw training data), model latent space (feature vectors produced by backbones), and prediction space (classification scores).

2. RESULTS

The results presented discuss an effective method of bridging the training data domain gap in §2.1, and domain gap measurements in §2.2.

2.1 Data Augmentation Impact on Model Generalization

The initial training dataset, validation dataset, and test dataset are small, with 869 real world (RW) and virtual world (VW) training instances (226 images), 140 validation instances (29 images), and 69 test instances (29 images) respectively. The test dataset only includes real images. The inclusion of augmented data is meant to ensure that test instances are within the distribution of the training data. This is visualized by extracting the feature vectors of the dataset from the YOLOv8 backbone, performing dimensionality reduction with Principal Component Analysis[13] (PCA) on the training dataset extended with augmented data, reducing the dimensionality of the test data using the PCA model fit on the extended training data, then plotting the results as shown in Fig.

2. PCA maximizes the representation of the variance of the data,[13] and therefore will indicate if the augmented data is expected to add to the overall variance of the training dataset.

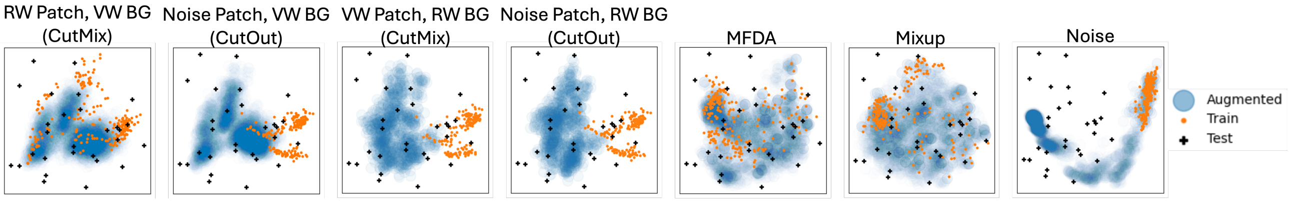

Figure 2. Dimensionality reduction using PCA of the baseline training data with augmented data types: real world (RW) patch on virtual world (VW) background (BG), noise patch on virtual world (VW) background (BG), virtual world (VW) patch on real world (RW) background, mixed feature data augmentation (MFDA), Mixup,[10] and noise.[11]

The PCA results qualitatively show that the Mixup augmentation method most uniformly covers the space of the augmented data, the baseline training dataset, and the test dataset, while the space is least well covered by data augmented with patches. This motivates the intuition that YOLOv8 models trained with Mixup augmented data may perform best, and models trained with CutMix and CutOut augmented data may not perform well.

This intuition is tested in Figure 3, where the results of training a YOLOv8[7] model with different kinds of augmented data are presented. Each data point represents the performance of a model retrained from frozen backbone weights using increasing numbers of augmented images, and then evaluated using the same test set of 69 test instances. The baseline F1 score from training exclusively on real world (RW) and virtual world (VW) is shown as a dashed line.

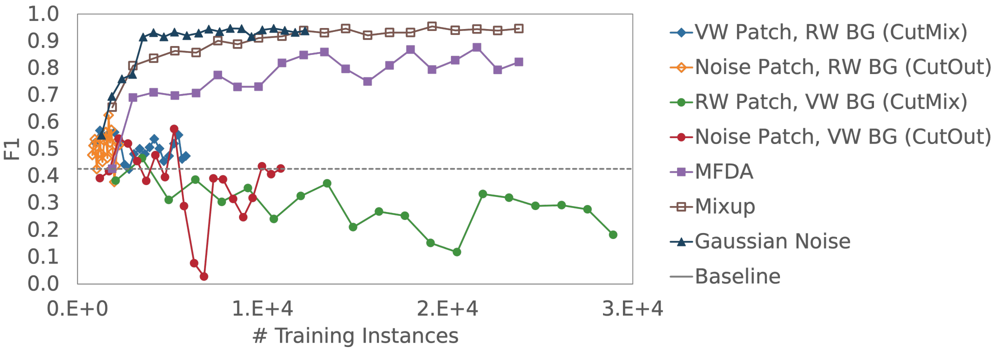

Figure 3. Results of transfer-learning training a YOLOv8 model with increasing amounts of differently augmented training data. The test set used to calculate the F1 score is held constant for all models.

Of the seven different sets of augmented data, the model generalized best to the unseen test set with white noise[11] augmentation and Mixup[10] augmentation; with the inclusion of error bars (not shown), we consider these two augmentation methods to be equivalently high performing. The models trained with CutMix and CutOut augmentation methods do not perform well, with the augmentation of synthetic virtual world (VW) images with noise patches or real world (RW) object instance patch showing a significant decrease in performance compared to all other methods. The models trained with increasing amounts of mixed feature data augmented (MFDA) images show significant improvement over the baseline, but do not perform as well as the noise and Mixup models.

This leads to our first result: **Models trained with pixel-level augmentation outperformed models trained with patch-level augmentation.** This suggests that regularization methods such as Mixup which are independent of pixel distances[14] may be more effective in bridging domain gaps. This is discussed further in §3.1.

2.2 Domain Gap Measurements

We now attempt to determine if there is a domain gap measurement on the training data which predicts the generalization of the model to the dataset outliers in the test data. Using Eq. 1 as our framework for evaluating the gap, the self-similarity of the dataset was determined using the measures summarized in Table 1.

Table 1. Summary of relationships $m(x, y)$ used in Eq. 1 to determine similarity. D_{KL} is the KL-Divergence.[15]

Measure $m(x, y)$	Equation	Evaluation Space
Mean Square Error[15] (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$	image, latent
Structural Similarity Index[16] (SSIM)	$luminance(x, y)\ \cdot contrast(x, y)\ \cdot\ structure(x, y)$	image
Binary Cross Entropy[17] (BCE)	$-(y\log(x) + (1 - y)\log(1 - x))$	image, latent
Jensen Shannon Divergence[18] (JSD)	$\frac{1}{2}D_{KL}\left(x\mid\frac{x+y}{2}\right) + \frac{1}{2}D_{KL}\left(y\mid\frac{x+y}{2}\right)$	image, latent
Fréchet Inception Distance[19] (FID)	$\left(inf_{\gamma\in\Gamma(\mu,\nu)}\int \Vert x - y\Vert^2 d\gamma(x, y)\right)^{1/2}$	latent
Kernel Inception Distance[20] (KID)	$MaximumMeanDiscrepancy(x, y)^2$	latent
Inception Score[21] (IS)	$exp\left(D_{KL}\left(x, y\right)\right)$	prediction

After evaluating the metrics in various domains, the best metric for each domain is shown in Fig. 4 based on the criteria discussed in §4.4. Each data series plotted in Fig. 4 represents the movement of the training data distribution in the image, latent, or prediction space for increasing amounts of augmented data. This is accomplished by plotting how the mean value and the standard deviation of the mean value for a given metric vary; the arrows point in the direction of increasing amounts of augmented data and show which direction the augmented data is trending.

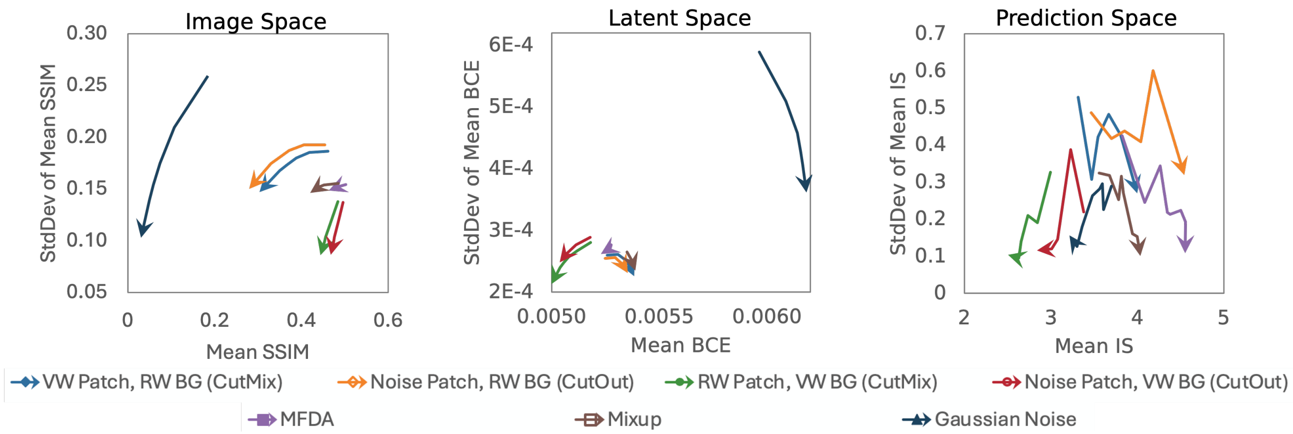

Figure 4. Best domain gap measurements for the image space (left) using Statistical Similarity Index, latent space (center) using Binary Cross Entropy, and prediction space (right) using the Inception Score.

Based on the criteria for a good measure of distribution gap (described in §4.4) and out of the measures in Table 1, SSIM[16] is the best predictor of future model performance in the image space due to its ordering of best performing augmentation "Gaussian Noise" in the upper left corner (low mean, high standard deviation) and worst performing augmentation method "Noise Patch, VW BG" in the lower right corner (low standard deviation, high mean). For the latent space, the best predictor of model performance on the test set was Binary Cross Entropy[17] (BCE), with poorly performing augmentation methods moving in the direction of decreased mean BCE value, and better performing augmentation methods moving in the direction of increased mean BCE value. In the prediction space, the Inception Score[21] (IS) was the only predictor of future model performance examined in this work, and does not predict model performance.

This leads to our second result: **A model's future performance was best predicted in the image space using a derived measure (SSIM), and in the latent space using a probabilistic measure (BCE).** The prediction space of the augmented training data was not found to be indicative of future model performance. The implications of the different measures are discussed further in §3.2.

3. DISCUSSION

3.1 Data Augmentation as Regularization Technique

Data augmentation–whether through synthetic images or augmented images–is a source of invariance and stochasticity in the data,[22] where the randomization of domain-dependent features is known to improve generalization to out-of-distribution instances.[23] While the task of training on synthetic data and predicting on real data is framed as a domain generalization or domain shift task,[24] mixed sample data augmentation (MSDA) is typically understood in the context of regularization.[8, 14]

The type of regularization contributed by patch-level (e.g. CutMix,[8] CutOut[9]) and pixel-level methods (e.g. Mixup,[10] Gaussian Noise[11]) was analyzed by Park et al.:[14] pixel-level methods were found to regularize the model Hessian and gradient without dependence on the distance from one pixel to the next, while the patch-level methods were found vary in the strength of the regularization. For the real and synthetic data gap presented in this work, we interpret the success of pixel-level augmentation over patch-level augmentation to indicate that all features in the real world (RW) and virtual world (VW) images required equal regularization; that is, no RW or VW feature required more regularization than any other feature. This suggests that the underlying domain gap between the RW and VW distributions is intrinsic to the data, and not driven by a single feature of the dataset (e.g., an aspect ratio present in one distribution but not the other), and explains the divergence in F1 scores between patch-level and pixel-level augmentation methods in Fig. 3.

The improvement of the model given the pure noise augmentation is less surprising than it may appear. The original dataset is small, and the model is therefore prone to overfitting on the single class of the dataset. The addition of pure noise prevents the model from converging prematurely to a local minimum by adding stochasticity to the optimization path.[11, 22] This directly motivates the superior performance of the noise-augmented models in Fig. 3; in this study the addition of noise was a pixel-level augmentation (see Fig. 5), and the use of soft labels with increasing noise effectively improved the performance of the model.

3.2 Implications of Similarity Measures

Two complimentary interpretations of each image in the training dataset exist. In the first interpretation, each image represents a vector in a high-dimensional space, with each pixel representing a vector component. This motivates the use of distance based measures, including MSE,[15] SSIM,[16] the cosine similarity[25] and the Mahalanobis distance.[26] In the second interpretation, each pixel represents a sample from an unknown probability distribution, which in turn motivates entropic and divergence measures such as the Binary Cross Entropy[17] (BCE), KL-Divergence,[15] and the Jensen-Shannon Divergence.[18] An equivalent way of expressing these two interpretations is that some similarity measures for images only assume a generalized Euclidean metric, while other similarity measures assume the information-specific Fisher metric.[27]

The SSIM index[16] is a derived metric that relates the weighted mean and standard deviation of pixel values between two images; it assumes that the image itself is the distribution to be measured. The structural aspect of the measure comes from the application of a two-dimensional scanning window with a Gaussian kernel during the calculation of the SSIM value.[16] The decrease in mean and standard deviation SSIM with increasing amounts of augmented data suggests that on average, the images in the dataset are becoming more similar through augmentation. The inclusion of Gaussian noise in the dataset is most effective at achieving the increased average similarity, which is unsurprising as all images are increasingly being augmented by drawing from a third, shared Gaussian distribution. What is surprising is that the use of Mixup and MFDA to augment the dataset caused the measured similarity to change the least (the arrows for these two methods are shortest in Fig. 4, even though there are nearly three times as many training instances as the noise augmented training datasets and the number of images in the noise, MFDA, and Mixup datasets are identical). The SSIM results in Fig. 4 fail to motivate why the pixel-level augmentation methods outperform the patch-level augmentation methods. However, the increased predictive value of SSIM over MSE suggests that considering each pixel to be an independent vector component is less useful than considering groups of pixels at once, even given the pixel-level augmentation techniques.

In the latent space, BCE assumes that each component of the feature vector extracted from the YOLOv8 backbone is a measure of the probability distribution learned by the model. The mean BCE increases for the effective dataset augmentation methods Gaussian noise, Mixup, VW Patch on RW BG (CutMix), and Noise Patch on RW BG (CutOut), all of which improve over the baseline in Fig. 3. The inclusion of MFDA data does not affect the training data distribution in the latent space as measured by BCE. The effect of including patch-level augmentation on virtual world synthetic images is to decrease the mean BCE value of the dataset. These results represent the most predictive measure of model performance reported in this work, where the increase or decrease in mean BCE directly correlates with increase or decrease in the model's F1 score (excepting the MFDA results). The effectiveness of the Gaussian noise can again be attributed to the inclusion of a third distribution in the training data, which effectively increases the entropy.

The result of evaluating the augmented training data using the Jensen-Shannon Divergence[18] is qualitatively similar to the results presented for BCE in Fig. 4, but less sensitive to increasing amounts of augmented data and fails to correctly predict the high performance of the Mixup augmented data.

There is no good result reported for the prediction space; the result for the Inception Score[21] (IS) in Fig. 4 has no predictive value for the F1 scores reported in Fig. 3.

3.3 Data Augmentation Effect on Underlying Distribution

Why some data augmentation effects work for some tasks and not others is not well explained.[14] The results in this paper suggest that such an explanation might be motivated by understanding how adding the augmented data changes the training distribution.

Although the best choice of augmentation method for a given task has been explained by the regularization effect on the model (as discussed in §3.1), the visualizations of the data in Fig. 2 suggest that the augmentation data effects may also be understood separately from the model, as the PCA result is independent of the YOLOv8 model results. The four kinds of patch-level augmentation form clusters, while the MFDA and Mixup augmented data is distributed more evenly across the PCA space. Comparing only these six types of data augmentation would lead to the intuition that data augmentation that evenly covers the PCA space shared by the training and test data distributions effectively closes the domain gap and will result in improved model performance. This intuition, while not proven false by the results shown in Fig. 3, is shown to be naive by the inclusion of the PCA result for the noise-augmented data: most of the testing data images continue to be outliers compared to the training and augmented data distributions in this visualization.

An additional motivation for nuanced analysis of the source of the effectiveness for different augmentation methods is found in Fig. 4. Plotting the mean against the standard deviation of the mean shows how the first two moments of a distribution change with respect to each other. For the image space and latent space results presented in Fig. 4, arrows which are close together suggest a shared distribution used to augment the data because the underlying data distribution is changed in the same way: the real world (RW) background (BG) augmented data consistently cluster together, as do the virtual world (VW) background (BG) data and the pixel-level augmentation Mixup and MFDA data. The grouping of the arrows may indicate that different kinds of randomness[28] are present in the different augmentation techniques; this could be confirmed by analyzing changes in the heaviness of the tails of the distributions for higher ordered moments.

4. METHODS

4.1 Baseline Dataset

The original dataset consists of a single class, ships, and was constructed after previously reported methods.[5] Real world instances were sampled from the Singapore Maritime Dataset[29] and internet videos of ship spotting.[5] The real world instances were combined with ship data created from virtual scenes fabricated in-house.[5] These images were used to create a baseline dataset containing 45 real world images and 180 virtual world images with a 20:80 real:synthetic ratio.

4.2 Data Augmentation

Seven different kinds of augmented data were prepared for this study. The first four kinds (real world (RW) object patches on virtual world (VW) backgrounds, VW object patches on RW backgrounds, noise patches on VW backgrounds, noise patches on RW backgrounds) are versions of CutMix[8] and CutOut[9] respectively, and were prepared after previously reported methods[5] so that each object patch had a twin noise patch on the same background in the same location. The fifth kind of augmented data was prepared using Mixup,[10] where the x_i and x_j images are always chosen so that one is synthetic and the other is real. This ensures that each augmented image is always a mix of real and synthetic pixels. Soft labels are introduced for the augmented images,[10] which directly affects the training of the classifier. The sixth kind of data begins with either a real or synthetic image, then increasingly adds Gaussian noise with soft labels until only a noise image remains, similar to the method presented by Zada et al.[11]

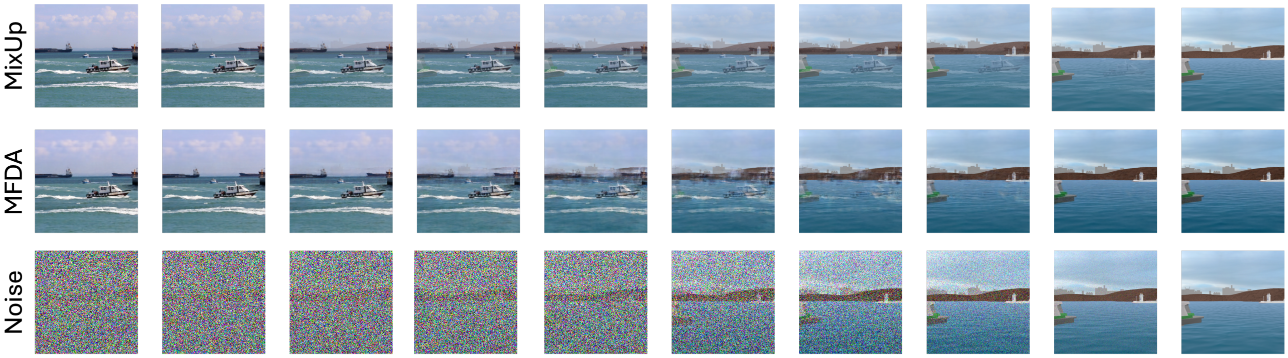

Figure 5. Augmented data using Mixup, Mixed Feature Data Augmentation (MFDA), and Noise. For MFDA and Mixup, the farthest left image is a real world example from the Singapore Maritime Dataset,[29] and the farthest right image is a virtual world example created in-house.

The seventh kind of data is a variation of Manifold Mixup,[12] where latent features are interpolated and soft labels are used to flatten decision boundaries. Manifold Mixup occurs within the model layers.[12] Here we use a Variational Autoencoder trained on the baseline dataset of real world and virtual world images as a surrogate latent space for the YOLOv8 model, interpolate between real world and virtual world image latent feature vectors, then convert the interpolated feature vectors to images using the VAE decoder.[30]

For Mixup, noise augmented data, and MFDA data, eight intermediate images were created for every interpolation between image x_i and image x_j, creating sequences of ten images. For the noise augmented data, this results in 10% of augmented images being pure noise, as shown in Fig. 5.

4.3 Model and Training

The Ultralytics YOLOv8[7] model was trained using transfer learning, where the backbone was initialized using the default weights trained on MSCOCO.[31] Training a model with short bursts of augmented data was previously found to be successful,[6,32] and this knowledge was leveraged by training the model on the baseline and augmented datasets for 90 epochs with all other hyperparameters left at the default. The F1 scores were calculated with a confidence threshold of 0.5.

4.4 Domain Gap Metric Criteria

We propose that a good measure for domain gaps should meet the criteria discussed by Deng et al.,[33] namely that the measure **(1)** be easy to calculate (low computational complexity), **(2)** correlate the performance of the model on test data with the size of the gap in the training data (directional and predictive), **(3)** provide insight into the source of the gap (enhance model explainability), and **(4)** be sensitive to changes in the data distributions as efforts to close the domain gap are enacted.

4.5 Calculation of Similarity Measures

The Inception Score[21] is calculated in the prediction space using the PyTorch implementation.[34] The Fréchet Inception Distance[19] (FID), Kernel Inception Distance[20] (KID), and Jensen-Shannon Divergence[18] (JSD) calculations were implemented from the *pytorch-fid: FID Score for PyTorch* repository.[35] The FID metric was calculated using the second max pooling features (192) due to the small number of images in the training dataset. FID is criticized[36] because the features in InceptionNet are not representative of generative model features; feature representation mismatch could explain the failure of FID as a measure to predict the F1 scores of YOLOv8. KID is presented as an improvement on FID since FID has a Gaussian assumption, which KID removes.[20] The JSD was used over KL-divergence[15] because KL-divergence is undefined for values of zero, and zeros were present in both the image space and the latent space data.

5. SUMMARY

The augmentation of training data with pixel-level augmentation (Gaussian noise and Mixup) is found to be the most effective method for bridging a domain gap between real and synthetic images, followed by pixel-level augmentation techniques. The domain gap between the real and synthetic images is best indicated using a probabilistic measure (Binary Cross Entropy) in the latent space, where an increase in the mean Binary Cross Entropy for all pairwise comparisons in a dataset is indicative of increased model equivariance. This is explained through the regularization effects on the model, where pixel-level effects regularize all features equally, regardless of the distance between pixels. An observed correlation between the distribution source for augmentation and the covariance of the mean and standard deviation of the similarity measure is hypothesized to be due to different kinds of randomness, and this can be further explored by examining higher ordered moments of the similarity measure distribution for pairwise comparisons of an augmented dataset.

ACKNOWLEDGMENTS

We gratefully acknowledge the contributions of Sam Brunes, Will Bickel and Jasmine Martin in preparing the synthetic virtual world data, Che Bin Mike Foo, Adithya Kandru, and Siddhant Chunchuwar for their contributions to data preprocessing, and Kham Lin Nom in collecting results for this work. This work was funded by the US Department of Defense (Contract W52P1J2093009), NSWC-CRANE N00164-24-1-1001, and US Army Research Lab Congressional Add-on (CACI International W15P7T-19-D-0157).

REFERENCES

[1] Nowruzi, F. E., Kapoor, P., Kolhatkar, D., Hassanat, F. A., Laganiere, R., and Rebut, J., "How much real data do we actually need: Analyzing object detection performance using synthetic and real data," (2019).

[2] Kvinge, H., Emerson, T. H., Jorgenson, G., Vasquez, S., Doster, T., and Lew, J. D., "In what ways are deep neural networks invariant and how should we measure this?," (2022).

[3] Deng, W., Gould, S., and Zheng, L., "On the strong correlation between model invariance and generalization," *Advances in Neural Information Processing Systems* **35**, 28052–28067 (2022).

[4] Blum-Smith, B. and Villar, S., "Machine learning and invariant theory," (2023).

[5] Dale, A. S., Christopher, L., Reindl, W., Sanchez, E., Brunes, S., Bickel, W., Martin, J., and William, A., "All patched up: effective integration of real and synthetic features into a single image for object detection," in [*2023 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*], 1–7 (2023).

[6] He, Z., Xie, L., Chen, X., Zhang, Y., Wang, Y., and Tian, Q., "Data augmentation revisited: Rethinking the distribution gap between clean and augmented data," (2019).

[7] Jocher, G., Chaurasia, A., and Qiu, J., "Ultralytics YOLO," (Jan. 2023).

[8] Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y., "Cutmix: Regularization strategy to train strong classifiers with localizable features," in [*Proceedings of the IEEE/CVF international conference on computer vision*], 6023–6032 (2019).

[9] DeVries, T. and Taylor, G. W., "Improved regularization of convolutional neural networks with cutout," (2017).

[10] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D., "mixup: Beyond empirical risk minimization," (2017).

[11] Zada, S., Benou, I., and Irani, M., "Pure noise to the rescue of insufficient data: Improving imbalanced classification by training on random noise images," in [*International Conference on Machine Learning*], 25817–25833, PMLR (2022).

[12] Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y., "Manifold mixup: Better representations by interpolating hidden states," in [*International conference on machine learning*], 6438–6447, PMLR (2019).

[13] Abdi, H. and Williams, L. J., "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics* **2**(4), 433–459 (2010).

[14] Park, C., Yun, S., and Chun, S., "A unified analysis of mixed sample data augmentation: A loss function perspective," *Advances in Neural Information Processing Systems* **35**, 35504–35518 (2022).

[15] Asperti, A. and Trentin, M., "Balancing reconstruction error and kullback-leibler divergence in variational autoencoders," *IEEE Access* **8**, 199440–199448 (2020).

[16] Bakurov, I., Buzzelli, M., Schettini, R., Castelli, M., and Vanneschi, L., "Structural similarity index (ssim) revisited: A data-driven approach," *Expert Systems with Applications* **189**, 116087 (2022).

[17] Creswell, A., Arulkumaran, K., and Bharath, A. A., "On denoising autoencoders trained to minimise binary cross-entropy," (2017).

[18] Lin, J., "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory* **37**(1), 145–151 (1991).

[19] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S., "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems* **30** (2017).

[20] Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A., "Demystifying mmd gans," (2018).

[21] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., "Improved techniques for training gans," (2016).

[22] Geiping, J., Goldblum, M., Somepalli, G., Shwartz-Ziv, R., Goldstein, T., and Wilson, A. G., "How much data are augmentations worth? an investigation into scaling laws, invariance, and implicit regularization," (2022).

[23] Gao, I., Sagawa, S., Koh, P. W., Hashimoto, T., and Liang, P., "Out-of-domain robustness via targeted augmentations," in [*Proceedings of the 40th International Conference on Machine Learning*], Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., eds., *Proceedings of Machine Learning Research* **202**, 10800–10834, PMLR (23–29 Jul 2023).

[24] Xu, Q., Zhang, R., Fan, Z., Wang, Y., Wu, Y.-Y., and Zhang, Y., "Fourier-based augmentation with applications to domain generalization," *Pattern Recognition* **139**, 109474 (2023).

[25] Xia, P., Zhang, L., and Li, F., "Learning similarity with cosine similarity ensemble," *Information sciences* **307**, 39–52 (2015).

[26] Ghorbani, H., "Mahalanobis distance and its application for detecting multivariate outliers," (2019).

[27] Masi, M., "Generalized information-entropy measures and fisher information," (2006).

[28] Mandelbrot, B. B. and Mandelbrot, B. B., [*States of randomness from mild to wild, and concentration from the short to the long run*], Springer (1997).

[29] Prasad, D. K., Rajan, D., Rachmawati, L., Rajabally, E., and Quek, C., "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Transactions on Intelligent Transportation Systems* **18**(8), 1993–2016 (2017).

[30] Dale, A. S. and Christopyer, L., "Direct adversarial latent estimation to evaluate decision boundary complexity in black box models," (2024).

[31] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., "Microsoft coco: Common objects in context," in [*Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*], 740–755, Springer (2014).

[32] Achille, A., Rovere, M., and Soatto, S., "Critical learning periods in deep neural networks," (2017).

[33] Deng, A. and Shi, X., "Data-driven metric development for online controlled experiments: Seven lessons learned," in [*Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*], 77–86 (2016).

[34] Obukhov, A., Seitzer, M., Wu, P.-W., Zhydenko, S., Kyl, J., and Lin, E. Y.-J., "High-fidelity performance metrics for generative models in pytorch," (2020). Version: 0.3.0, DOI: 10.5281/zenodo.4957738.

[35] Seitzer, M., "pytorch-fid: FID Score for PyTorch." https://github.com/mseitzer/pytorch-fid (August 2020). Version 0.3.0.

[36] Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., and Kumar, S., "Rethinking fid: Towards a better evaluation metric for image generation," (2023).