Effect of GAN-Augmented Data on CNN Training

EE583 Term Project Report

Yunus Emre Tüysüz
Electrical and Electronics Engineering
Middle East Technical University
Ankara, Turkey
jwarsemre@gmail.com

***Abstract*—Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs) are pivotal in modern image classification and data augmentation. This study explores the integration of GANs and CNNs to address challenges posed by datasets of varying complexity and scarcity, including ultrasound, MNIST, and CIFAR-10. GANs are employed to generate synthetic data, augmenting CNN training, particularly for small or complex datasets. Experiments compare the performance of GAN-augmented CNNs, standalone CNNs, and traditional classifiers such as SVMs, highlighting the advantages of GAN augmentation in enhancing classification accuracy. Furthermore, the study examines the diminishing impact of GANs as dataset size increases and discusses challenges such as mode collapse and computational costs. These findings enhance our understanding of the role of GANs in improving CNN-based image classification and data augmentation.**

Keywords—generative adversarial networks, convolutional neural networks, data augmentation, auxiliary classifier, support vector machines

I. Introduction

This report follows the methodology presented in the article *Ultrasound Image Classification Using ACGAN with a Small Training Dataset* [1]. Additional insights are drawn from *Conditional Activation GAN: Improved Auxiliary Classifier GAN* [2] and *CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection* [3]. The primary objective is to evaluate the effectiveness of GAN-augmented datasets in improving image classification performance.

The report begins by presenting the results from the referenced article, followed by an analysis of experiments conducted on additional datasets. These include tests with CNN-only models, GAN-augmented datasets, and support vector machines (SVMs) implemented in MATLAB.

Given the limitations of the initial dataset, larger datasets are also analyzed to assess the impact of data diversity on training and classification accuracy. This approach aims to thoroughly evaluate the role of GAN augmentation in ultrasound image classification and its general applicability to other datasets.

II. Theory

In this section, the theory behind GANs and the process of creating GAN-augmented datasets will be examined. Additionally, the model and algorithm will be explained.

A. Generative Adversarial Networks

Introduced by Ian Goodfellow in 2014 [4], GANs consist of two neural networks: the **Generator** and the **Discriminator**, which engage in a zero-sum game framework.

- **Generator**: Creates images (or other data types like signals) that resemble real data, starting from random noise.
- **Discriminator**: Evaluates input data to determine whether it is real or generated (fake).

Through adversarial training, both networks improve iteratively: the Generator learns to produce increasingly realistic data, while the Discriminator becomes better at distinguishing between real and generated data.

The architecture is shown in Figure 1. While the generator learns to create fake images, the discriminator learns to determine whether the input is real or fake. Through this process, the generator mimics the distribution of the input samples, which aids in data augmentation in our case and allows the discriminator to be exposed to a wider variety of samples.

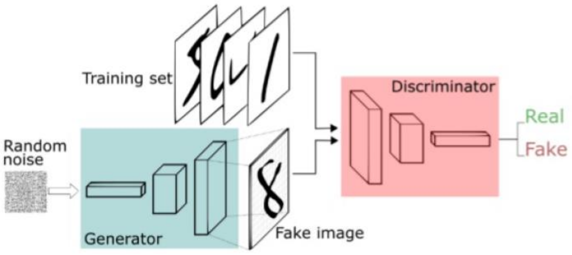


Figure 1 GAN architecture

B. Auxiliary Classifier GAN

Auxiliary Classifier GANs (ACGANs) enhance the quality and diversity of generated images by conditioning on class labels. This targeted generation process is particularly useful for addressing the challenges of limited or imbalanced datasets. By creating diverse, class-specific synthetic samples, ACGANs help augment datasets, balance underrepresented classes, and improve the performance of classifiers. They are especially valuable in applications where obtaining labeled data is costly or difficult, such as medical imaging or rare event detection.

The diagram is shown in Figure 2. The generator takes the class label and noise as input and learns to create a fake image corresponding to the given class. The discriminator, on the other hand, learns both to determine whether an image is real or fake and to classify the image's class. Both networks are trained through backpropagation, and at the end of the process, the generator is capable of producing labeled data, while the discriminator becomes adept at distinguishing and classifying samples from the same class distribution.

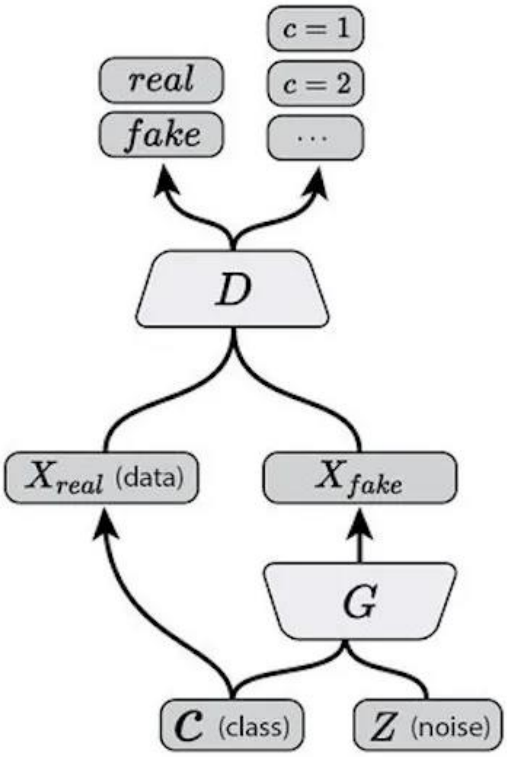


Figure 2 ACGAN architecture

C. Model

A similar architecture to that described in [2] is used. This is a DCGAN architecture consisting of convolutional layers, batch normalization, and leaky ReLUs for the discriminator, along with a fully connected layer for classification and real/fake detection. The generator employs transposed convolutions. The kernel size and padding are adjusted to accommodate the input image size, and the number of convolutional layers is also modified accordingly.

The overall architecture is shown in Figure 3. The generator creates an image from noise using transposed convolutions, while the discriminator takes an image as input, extracts features, and determines the class and whether the image is real or fake using convolutional layers.

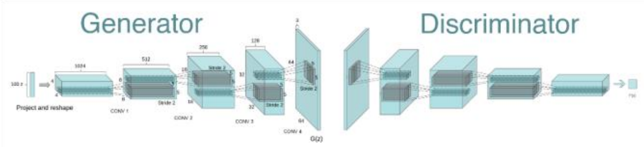


Figure 3 DCGAN architecture

For the CNN-only cases, the discriminator is used directly without the generator component. The discriminator functions as a standalone convolutional neural network, extracting features and performing classification tasks.

For the SVM implementation, the fitcecoc function is used. This function fits multiclass models by combining binary support vector machines (SVMs) or other classifiers using the Error-Correcting Output Codes (ECOC) framework, enabling effective classification in multiclass settings.

A more detailed version of the model can be found on the GitHub page in the Appendix. The code is inspired by the repository provided in [5].

III. Results and discussion

In this section, the results for the Ultrasound, MNIST, and CIFAR-10 datasets will be presented. For each dataset, the generated samples, confusion matrices, and accuracies will be provided. To maintain consistency, the parameters for each case will remain the same.

A. Ultrasound Classification

The Ultrasound dataset consists of 250 images divided into two categories: benign and malignant. Of these, 150 images belong to the malignant category, while the remaining 100 belong to the benign category. A total of 200 images are used for training, and 50 are reserved for testing. First, a GAN-augmented CNN is trained, followed by a standalone CNN. Additionally, the results from the referenced article are provided below.

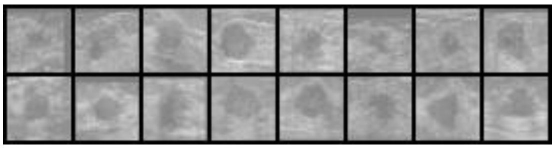


Figure 4 Real images

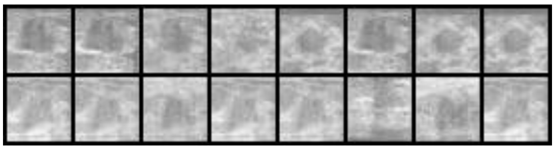


Figure 5 Fake images

Figures 4 and 5 compare real ultrasound images with GAN-generated ones. The fake images closely mimic the texture and patterns of the real ones, demonstrating the GAN’s effectiveness in learning and reproducing key features. This realistic augmentation enhances dataset diversity, potentially improving classification performance, though subtle artifacts may require validation to avoid affecting results. Figure 6 and 7 compares the training with and without GAN augmentation.

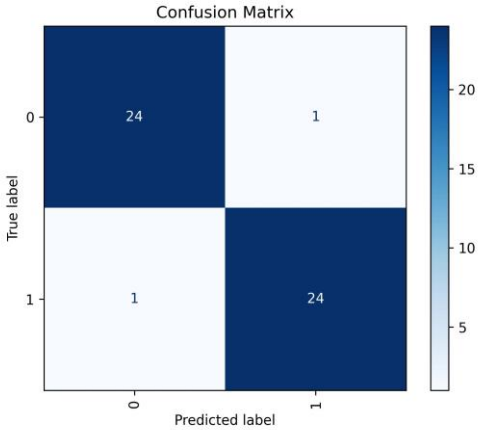


Figure 6 Confusion Matrix for GAN-augmented CNN

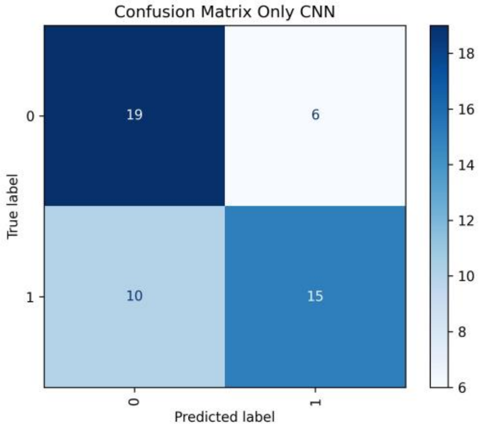


Figure 7 Confusion Matrix for standalone CNN

Also, the accuracies can be seen in the Table 1.

Table 1 Accuracies

Model	Accuracy
GAN-augmented CNN	96%
Standalone CNN	68%

Figures 6 and 7, along with Table 1, demonstrate the significant improvement in classification accuracy achieved by using a GAN-augmented CNN compared to a standalone CNN. The confusion matrix for the GAN-augmented CNN (Figure 6) shows near-perfect classification, with only one misclassification for each class, resulting in an accuracy of 96%. In contrast, the standalone CNN confusion matrix (Figure 7) highlights several misclassifications, especially for the malignant category, leading to a much lower accuracy of 68%. This improvement underscores the value of GAN-generated images in compensating for the dataset's small size, enhancing the model's ability to generalize. Additionally, even with a same training setup, such as a same model size, learning rate and minimal optimization adjustments, the GAN-augmented approach outperforms the standalone model by a wide margin, as evidenced in both the confusion matrices and the accuracy table.

The accuracies in the article can be seen in Table 2.

Table 2 Accuracies in article

Model	Accuracy
Proposed [1]	98.80%
Resnet-18	95.60%
VGG16	95.60%
VGG19	96.40%

The accuracies in Table 2 clearly show that the models in the referenced article perform significantly better than my CNN model, even though we used the same inputs. This difference could be attributed to their models being more optimized and better suited for the task, whereas my CNN architecture is larger and likely requires more data to train effectively. With a limited dataset, my model struggles to generalize as well as the referenced models, which might have more efficient designs or advanced training techniques that help them achieve higher accuracy. This highlights the importance of balancing model complexity with the available data size to ensure optimal learning. But my GAN augmented CNN works really well. For GAN I used the same parameters as them and result are quite similar.

B. MNIST classification

To better evaluate the effect of GAN, I also trained my model using a different dataset. Specifically, I experimented with the MNIST dataset, where I varied the input sizes to observe the model's behavior under different conditions. The goal was to assess how the models perform when the classification problem is relatively simple, as MNIST is a straightforward task compared to ultrasound image classification. By using this dataset, I aimed to isolate the impact of GAN-generated data and understand how the model's architecture responds when data complexity and difficulty are reduced. The fake images generated using only 5000 input images can be seen in the Figure 8

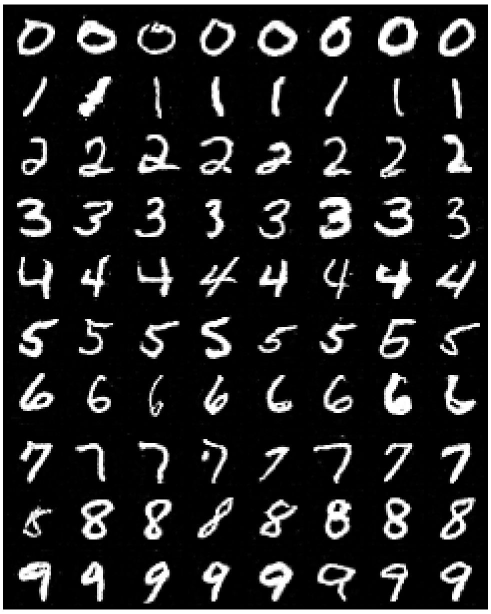


Figure 8 Fake images

From Figure 8, we can see that our generator works well. The fake images closely mimic the texture and patterns of the real ones. Figure 6 and 7 compares the training with and without GAN augmentation.

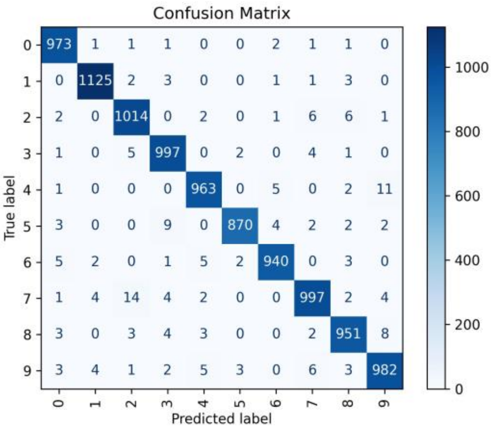


Figure 9 Confusion Matrix for GAN-augmented CNN

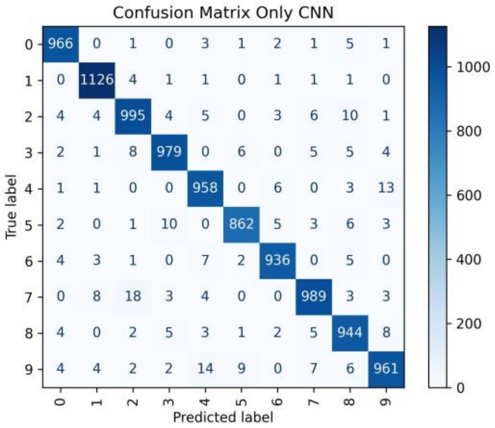


Figure 10 Confusion Matrix for standalone CNN

Figure 11 shows fake images generated using all the samples from the MNIST dataset.

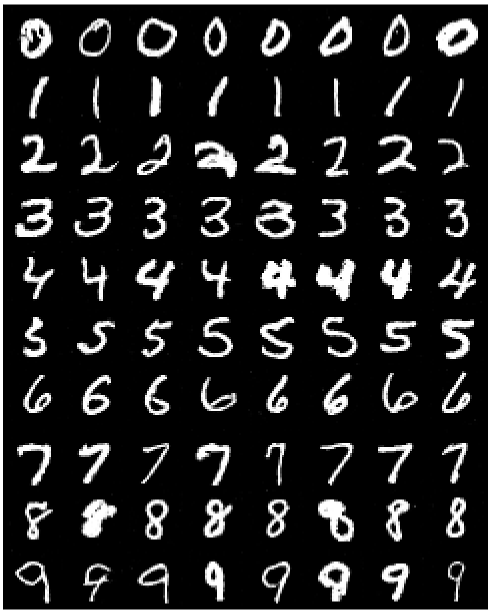


Figure 11 Fake images

When we compare Figures 8 and 11, we can see that the generator performs better for Figure 11. This also indicates that the generator requires larger datasets to accurately mimic real images.

Figure 12 and 13 shows the confusion matrix for the training with and without GAN augmentation.

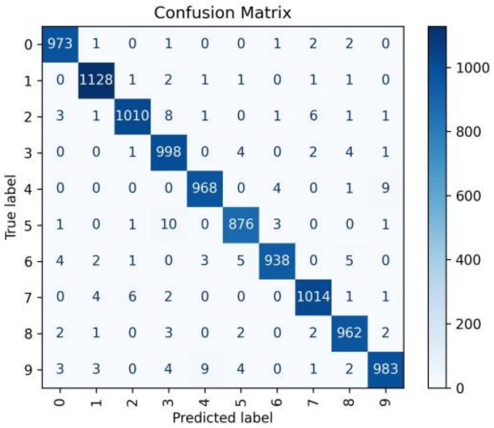
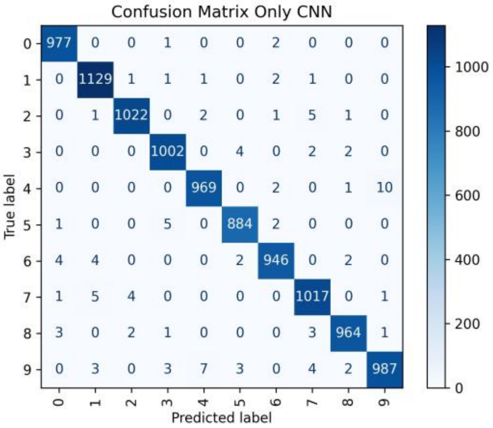


Figure 12 Confusion Matrix for GAN-augmented CNN



True label	0	1	2	3	4	5	6	7	8	9
0	977	0	0	1	0	0	2	0	0	0
1	0	1129	1	1	1	0	2	1	0	0
2	0	1	1022	0	2	0	1	5	1	0
3	0	0	0	1002	0	4	0	2	2	0
4	0	0	0	0	969	0	2	0	1	10
5	1	0	0	5	0	884	2	0	0	0
6	4	4	0	0	0	2	946	0	2	0
7	1	5	4	0	0	0	0	1017	0	1
8	3	0	2	1	0	0	0	3	964	1
9	0	3	0	3	7	3	0	4	2	987

Figure 13 Confusion Matrix for standalone CNN

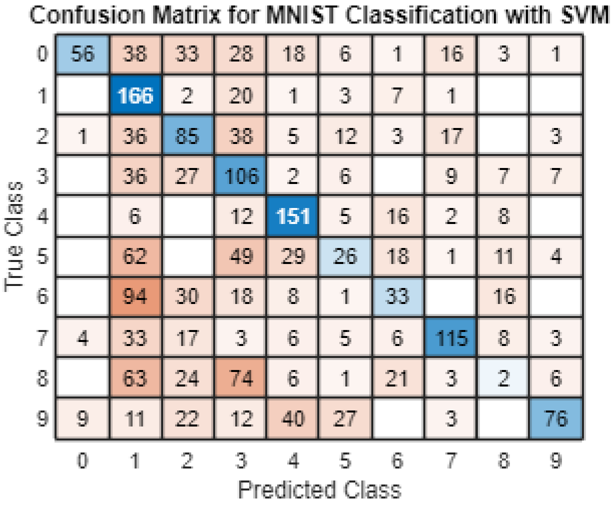
Also, the accuracies can be seen in Table 3.

Table 3 Accuracies

Model	Number of samples used for training	Accuracy
GAN-augmented CNN	5000	98.12%
Standalone CNN	5000	97.16%
GAN-augmented CNN	60000	98.50%
Standalone CNN	60000	98.97%

When we compare the results in Figures 9, 10, 12, and 13, along with Table 1, we observe that the effect of the GAN is relatively small, even when the number of samples is limited. This is primarily because the MNIST dataset is inherently easy to classify, allowing the standalone CNN to perform well without the need for GAN augmentation. The inputs in the MNIST dataset are clearly distinguishable by nature, unlike in the ultrasound dataset, where visual differentiation is much more challenging. Additionally, as the number of training samples increases, the standalone CNN achieves a similar accuracy to the GAN-augmented model, since both models use the same classifier architecture. This suggests that for simpler datasets like MNIST, where the features are easily separable, the advantage of GAN-generated data diminishes, highlighting the importance of dataset complexity in determining the effectiveness of augmentation techniques like GANs.

The results obtained from MATLAB using SVM are presented in Figure 14 as follows:



True Class	0	1	2	3	4	5	6	7	8	9
0	56	38	33	28	18	6	1	16	3	1
1		166	2	20	1	3	7	1		
2	1	36	85	38	5	12	3	17		3
3		36	27	106	2	6		9	7	7
4		6		12	151	5	16	2	8	
5		62		49	29	26	18	1	11	4
6		94	30	18	8	1	33		16	
7	4	33	17	3	6	5	6	115	8	3
8		63	24	74	6	1	21	3	2	6
9	9	11	22	12	40	27		3		76

Figure 14 Confusion Matrix for SVM

The accuracy achieved using SVM, as shown in Figure 14, is 40.8%. For this analysis, the digit Dataset was used, with 80% of the data allocated for training and 20% for testing. The input size is comparable to the 5,000 cases used in the CNN and GAN experiments. This result highlights that, even for a relatively simple classification task like MNIST, the SVM struggles to classify the digits effectively. The low accuracy demonstrates that traditional methods like SVM are insufficient for achieving satisfactory performance in such tasks. This underscores the need for state-of-the-art solutions, such as deep learning architectures, which are better equipped to handle the complexities and nuances of modern classification problems, even for seemingly straightforward datasets.

C. CIFAR10 Classification

After working with the complex ultrasound dataset and the simple MNIST dataset, I wanted to evaluate the results on the CIFAR-10 dataset. The drawback of the ultrasound data was its limitation to only two classes, while the MNIST dataset was relatively easy to classify. Since MNIST images are 28x28, I adjusted the kernel size in the model to accommodate the 32x32 CIFAR-10 images. I experimented with three different sample sizes: a very small subset, a medium-sized subset, and the entire dataset to analyze the full effect of the GAN. For the small number of inputs, the generated fake images are shown in Figure 15.

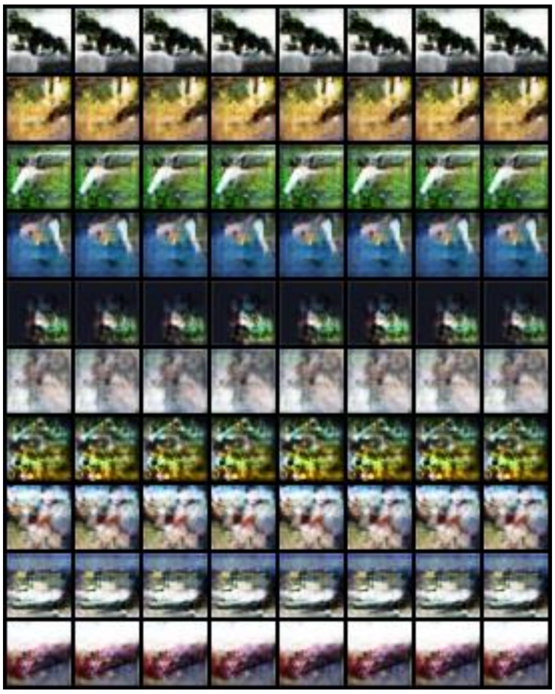
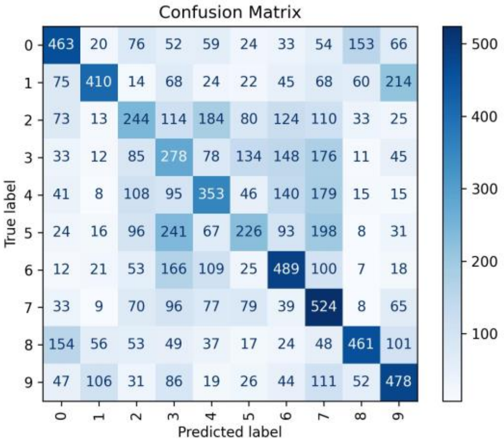


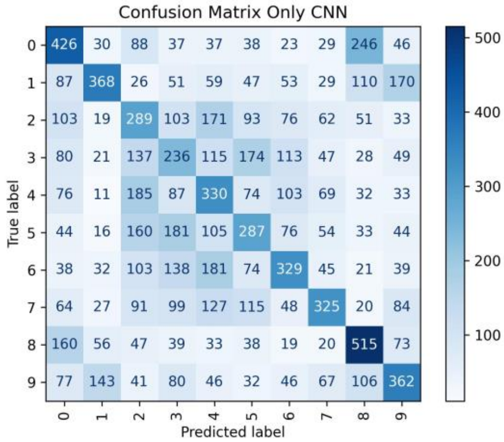
Figure 15 Fake images

The fake images are almost identical for each class, even when different noise is used for generating them. This indicates that our generator is not functioning properly, and our data augmentation is ineffective. The confusion matrices for the GAN-augmented CNN and the standalone CNN are shown in Figures 16 and 17, respectively.



True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	463	20	76	52	59	24	33	54	153	66
1	75	410	14	68	24	22	45	68	60	214
2	73	13	244	114	184	80	124	110	33	25
3	33	12	85	278	78	134	148	176	11	45
4	41	8	108	95	353	46	140	179	15	15
5	24	16	96	241	67	226	93	198	8	31
6	12	21	53	166	109	25	489	100	7	18
7	33	9	70	96	77	79	39	524	8	65
8	154	56	53	49	37	17	24	48	461	101
9	47	106	31	86	19	26	44	111	52	478

Figure 16 Confusion Matrix for GAN-augmented CNN



True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	426	30	88	37	37	38	23	29	246	46
1	87	368	26	51	59	47	53	29	110	170
2	103	19	289	103	171	93	76	62	51	33
3	80	21	137	236	115	174	113	47	28	49
4	76	11	185	87	330	74	103	69	32	33
5	44	16	160	181	105	287	76	54	33	44
6	38	32	103	138	181	74	329	45	21	39
7	64	27	91	99	127	115	48	325	20	84
8	160	56	47	39	33	38	19	20	515	73
9	77	143	41	80	46	32	46	67	106	362

Figure 17 Confusion Matrix for standalone CNN

For the medium number of input the Fake images can be seen in the Figure 18.

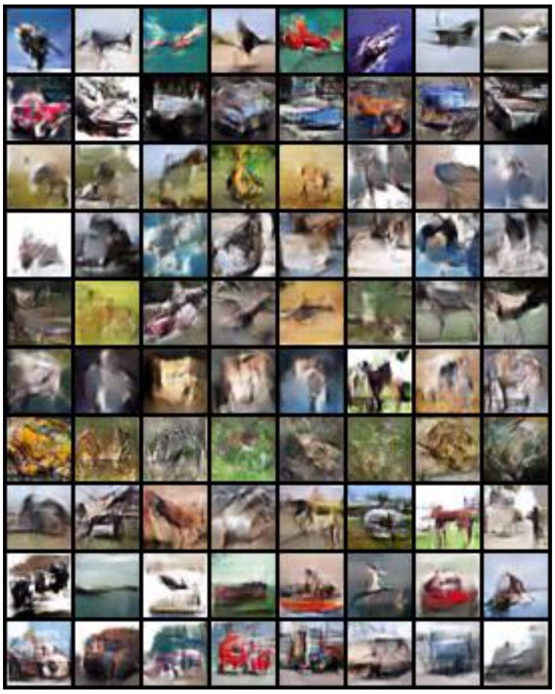
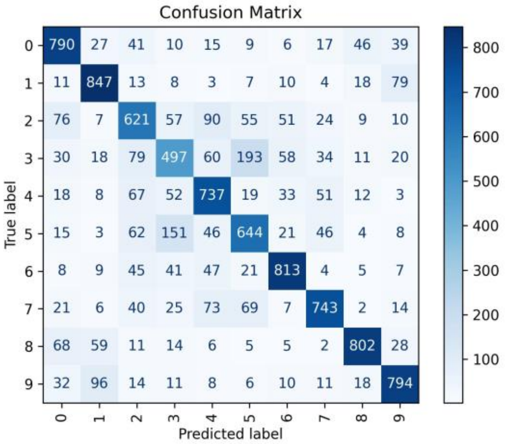


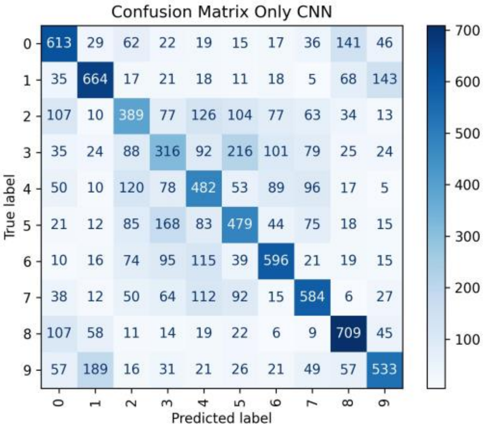
Figure 18 Fake images

Our generator started working better, which shows that even the generator needs sufficient data to function properly. For some pictures, even when the label is "dog," the generated image does not resemble a real dog. This indicates that the generator can trick the discriminator, and our classifier believes it is a dog image. The confusion matrices are shown in Figures 19 and 20.



	0	1	2	3	4	5	6	7	8	9
0	790	27	41	10	15	9	6	17	46	39
1	11	847	13	8	3	7	10	4	18	79
2	76	7	621	57	90	55	51	24	9	10
3	30	18	79	497	60	193	58	34	11	20
4	18	8	67	52	737	19	33	51	12	3
5	15	3	62	151	46	644	21	46	4	8
6	8	9	45	41	47	21	813	4	5	7
7	21	6	40	25	73	69	7	743	2	14
8	68	59	11	14	6	5	5	2	802	28
9	32	96	14	11	8	6	10	11	18	794

Figure 19 Confusion Matrix for GAN-augmented CNN



	0	1	2	3	4	5	6	7	8	9
0	613	29	62	22	19	15	17	36	141	46
1	35	664	17	21	18	11	18	5	68	143
2	107	10	389	77	126	104	77	63	34	13
3	35	24	88	316	92	216	101	79	25	24
4	50	10	120	78	482	53	89	96	17	5
5	21	12	85	168	83	479	44	75	18	15
6	10	16	74	95	115	39	596	21	19	15
7	38	12	50	64	112	92	15	584	6	27
8	107	58	11	14	19	22	6	9	709	45
9	57	189	16	31	21	26	21	49	57	533

Figure 20 Confusion Matrix for standalone CNN

For the entire data set used case Fake images can be seen in the Figure 21.

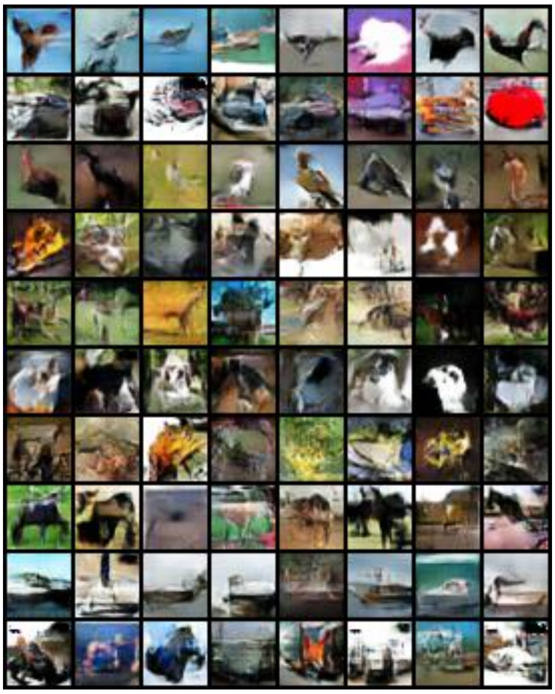
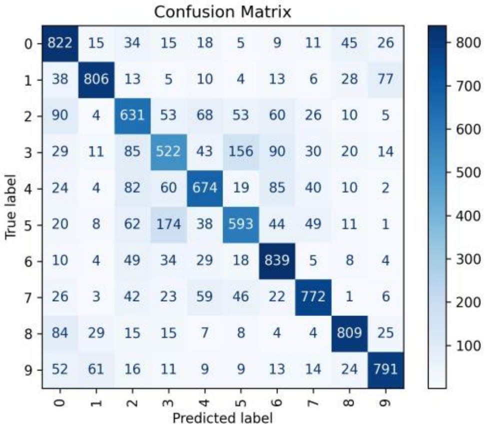


Figure 21 Fake images

The fake images closely resemble the real images when we use all the samples from the CIFAR-10 dataset. The confusion matrices are shown in Figures 12 and 23.



	0	1	2	3	4	5	6	7	8	9
0	822	15	34	15	18	5	9	11	45	26
1	38	806	13	5	10	4	13	6	28	77
2	90	4	631	53	68	53	60	26	10	5
3	29	11	85	522	43	156	90	30	20	14
4	24	4	82	60	674	19	85	40	10	2
5	20	8	62	174	38	593	44	49	11	1
6	10	4	49	34	29	18	839	5	8	4
7	26	3	42	23	59	46	22	772	1	6
8	84	29	15	15	7	8	4	4	809	25
9	52	61	16	11	9	9	13	14	24	791

Figure 22 Confusion Matrix for GAN-augmented CNN

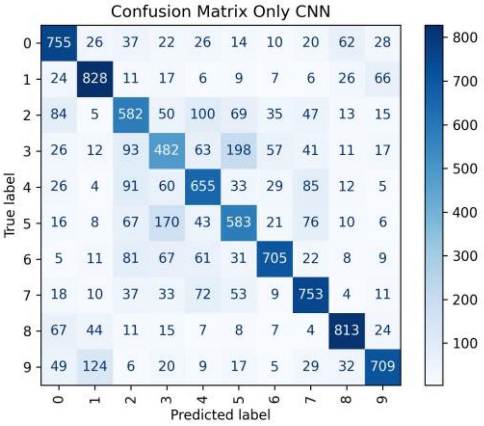


Figure 23 Confusion Matrix for standalone CNN

Accuracies can be seen in Table 4.

Table 4 Accuracies

Model	Number of samples used for training	Accuracy
GAN-augmented CNN	500	39.26%
Standalone CNN	500	34.67%
GAN-augmented CNN	5000	72.88%
Standalone CNN	5000	53.65%
GAN-augmented CNN	50000	72.59%
Standalone CNN	50000	67.55%

When we analyze Figures 16 and 17 (500-sample case), Figures 19 and 20 (5,000-sample case), and Figures 22 and 23 (50,000-sample case), two main observations can be made. First, the GAN requires a sufficient amount of data to learn and effectively mimic real images; its performance improves as the dataset size increases. Second, as the input dataset size grows larger, the importance of GAN augmentation diminishes. For example, in the 5,000-sample case, the GAN-augmented model achieves 20% higher accuracy compared to the standalone CNN, but in the 50,000-sample case, this difference reduces to only 5%. In the 500-sample case, the difference is also around 5%, indicating that the GAN is less effective with very small datasets. Using 5,000 samples strikes a good balance between computational efficiency and accuracy, delivering significant improvements. Additionally, it is evident that our CNN model saturates at around 72% accuracy, which is likely the best achievable with this specific model architecture.

The results obtained from MATLAB using SVM are presented in Figure 24 as follows:

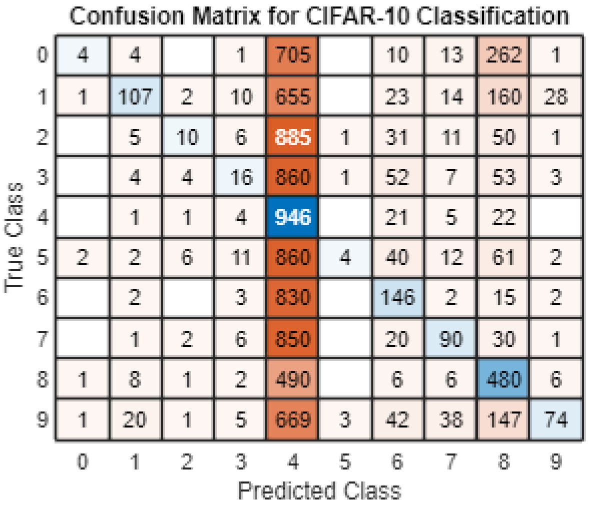


Figure 24 Confusion Matrix for SVM

The test accuracy is 18.7%, which is very poor. MATLAB's implementation appears to misclassify most of the classes as "4 (deer)," leading to an artificially inflated accuracy that is likely even worse. For this test, I used the same CIFAR-10 dataset as in the CNN and GAN experiments, downloading it and implementing the fitcecoc function. These results further demonstrate that traditional methods like MATLAB's default classifier are not suitable for complex datasets like CIFAR-10. This reinforces the need to use state-of-the-art solutions, such as deep learning models, which are far more capable of handling the complexities and nuances of modern image classification tasks.

IV. Conclusions

This study evaluated classification methods and data augmentation techniques across datasets including ultrasound, MNIST, and CIFAR-10, comparing traditional SVM approaches with modern deep learning methods like CNN and GAN-augmented CNNs.

The ultrasound dataset highlighted the benefits of GANs in small, complex datasets, where the GAN-augmented CNN outperformed the standalone CNN. For the simple MNIST dataset, where inputs are easily distinguishable, the GAN had a minimal impact as the CNN already achieved high accuracy. Traditional SVM methods struggled significantly, with only 40.8% accuracy on MNIST and 18.7% on CIFAR-10, demonstrating their limitations compared to deep learning models.

In CIFAR-10, sample size played a critical role. With smaller datasets, GAN augmentation provided significant accuracy boosts. However, the generator also requires sufficient samples to learn the input distribution effectively. When using very small datasets, the GAN fails to perform well. As the dataset size increased, the difference between GAN-augmented and standalone CNNs narrowed to 5%, demonstrating that the importance of GAN diminishes with larger datasets.

Overall, GAN-augmented CNNs are effective for improving accuracy in smaller, more complex datasets, while standalone CNNs excel with larger datasets. The results underscore the importance of using state-of-the-art methods like deep learning over traditional techniques for modern image classification tasks.

References

[1] S. Saha and N. Sheikh, "Ultrasound Image Classification using ACGAN with Small Training Dataset," *arXiv.org*, 2021. https://arxiv.org/abs/2102.01539.

[2] L. Hou, Q. Cao, H. Shen, S. Pan, X. Li, and X. Cheng, "Conditional GANs with Auxiliary Discriminative Classifier," *arXiv.org*, 2021. https://arxiv.org/abs/2107.10060.

[3] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "CovidGAN: Data Augmentation using Auxiliary Classifier GAN for Improved Covid-19 Detection," *IEEE Access*, pp. 1–1, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2994762.

[4] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," arXiv.org, Jun. 10, 2014. https://arxiv.org/abs/1406.2661

[5] B. Chao, "Anime-Generation," GitHub repository, https://github.com/bchao1/Anime-Generation.

Appendix

All codes are in https://github.com/jritzl/583_term_project