

# Exploiting synthetic images for real-world image recognition

Max Maton  
Delft University of Technology  
Delft, The Netherlands  
<https://in.maxmaton.nl/>

Miriam Huijser  
Aiir Innovations  
Amsterdam, The Netherlands  
<https://aiir.nl/>

Jan van Gemert  
Delft University of Technology  
Delft, The Netherlands  
<http://jvgemert.github.io/>

Osman Kayhan  
Delft University of Technology  
Delft, The Netherlands  
[o.s.kayhan@tudelft.nl](mailto:o.s.kayhan@tudelft.nl)

## Abstract

*This paper shows that the gap between synthetic and real-world image distributions can be closed by using GANs to convert the synthetic data to a dataset which has the same distribution as the real data. Training this GAN requires only a fraction of the dataset traditionally required to get a high classification accuracy. This converted data can subsequently be used to train a classifier with a higher accuracy than a classifier trained only on the real dataset.*

## 1. Introduction

Deep learning has revolutionised visual object recognition. Thanks to huge datasets and fast hardware (GPUs), current object recognition approaches have near-human accuracy. Because creating big datasets is often very expensive, people are starting to turn to synthetic images to augment their datasets. However, training networks on synthetic images may not achieve the desired accuracy due to a gap between synthetic and real image distributions. [8] Another development is the increased focus on Generative Adversarial Networks (GANs) to generate images that look similar to the images they were trained on. [6] In this paper the impact of the distribution gap between synthetic and real image distributions is decreased by using a GAN to modify the synthetic images to have the same distribution as real images. This technique is shown to be useful for inflating very small datasets to a level where they can be used to create accurate classifiers.

## 2. Related work

Artificial data can sometimes be used to train networks, i.e. a network trained on rendered images to segment im-

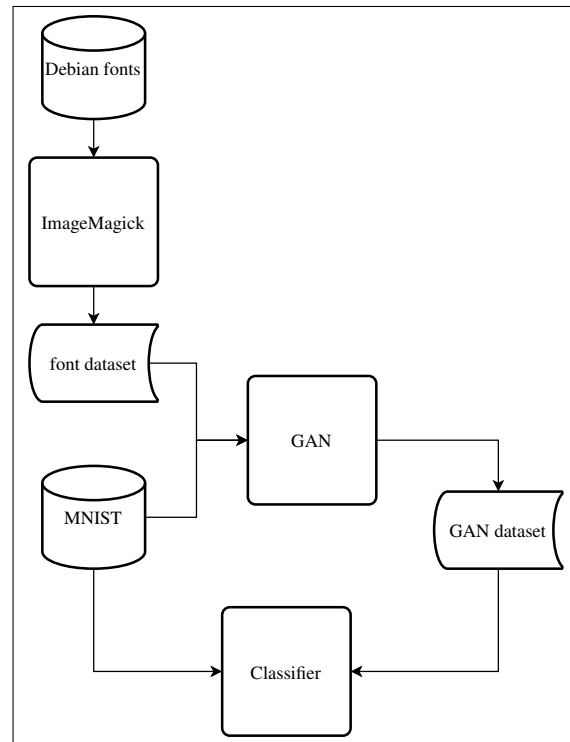


Figure 1. Technique used to convert synthetic dataset for use in training.

ages of indoor scenes [3], a network that recognises font characters trained on interpolated real samples [8] and a network trained with simulations to train control networks. [7] These networks are trained by creating a synthetic dataset that is as close as possible in statistical distribution as the real dataset. Generally this is very difficult or expensive to achieve.

A solution for the problem of creating synthetic datasets with a distribution close to a real dataset is domain adaptation. This is generally done using Generative Adversarial Networks [9, 2, 5] that are trained to create samples that are indistinguishable from images in the real dataset based on images in the crude synthetic dataset. We decided to perform this research on the GAN as described by Bousmalis et al [2] due to the available TensorFlow [1] implementation.

The problem of creating synthetic datasets with a distribution close to a real dataset can be solved by using domain adaptation. This is generally done using Generative Adversarial Networks [9, 2, 5] trained to create samples based on images in the crude synthetic dataset that are indistinguishable from images in the real dataset. We decided to perform this research on the GAN as described by Bousmalis et al [2] due to the available TensorFlow [1] implementation.

### 3. Inflating datasets using synthetic data and GANs

The basis of this research is the GAN that converts images from a synthetic dataset distribution to an image that appears to come from the distribution of a real dataset. This GAN consists of three parts. The first part is a generator network that does the image conversion. The second part of the GAN is a discriminator, which tries to distinguish between the output of the generator and images from the real dataset. The third part of the GAN is a classifier that predicts the label of images coming from either the real dataset, the generator or the synthetic dataset. All these networks are trained in parallel. The discriminator and classifier are trained to reduce the amount of misclassifications. The generator is trained to minimise the amount of pixels changed in the image, to minimise the loss in classification by the classifier and to try to fool the discriminator to classify the image as real.

To test the effects of inflating datasets when using this technique, we first trained the GAN with a real and synthetic dataset. We then use the trained GAN to create a new dataset and combine that dataset with the real dataset. This combined dataset is then used to train a second classifier.

#### 3.1. Experiments

We evaluated this technique on the MNIST dataset [4] together with a synthetic dataset generated by rendering open source font digits in multiple rotations for all number classes. The synthetic dataset contains samples from 149 fonts, with each digit rendered having 47 variations each with a distinct rotation between  $-47$  and  $46$  degrees. All images were normalised using the same algorithm used to normalise the MNIST samples, the result of which can be seen in figure 2. A random sample of 10,030 of those im-

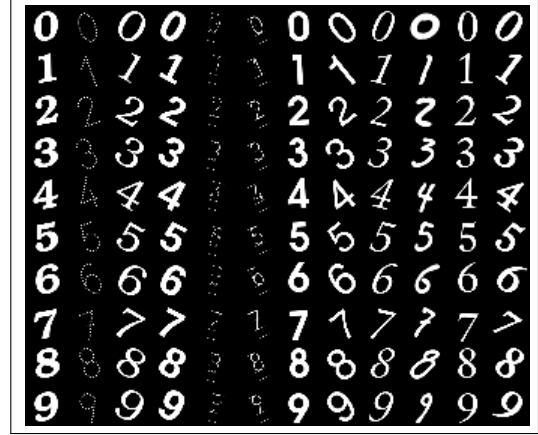


Figure 2. Example images of the MNIST font dataset

ages were put in a test set and the remaining 60,000 images were used as the synthetic dataset.

The GAN was modified to use 10% of the training data as a validation set instead of a constant 1,000 samples as this allowed for smaller sample sizes.

For multiple ratios  $r$ , where  $0 > r > 1$ , we created the following datasets:

$MNIST_{original}$   
 $60,000 \times r$  images from the original MNIST dataset.

$MNIST_{font}$   
 $60,000 \times (1-r)$  images from the synthetic font dataset.

We used these two datasets to train the GAN using 12 million training samples and subsequently ran the GAN on the  $MNIST_{font}$  dataset to create the  $MNIST_{GAN}$  dataset. This made the  $MNIST_{GAN}$  dataset exactly as big as  $MNIST_{font}$ .

With these datasets we trained five instances of the following classifiers:

$5 \times C-MNIST_{original}$   
Classifier only trained on  $MNIST_{original}$

$5 \times C-MNIST_{font}$   
Classifier only trained on  $MNIST_{font}$

$5 \times C-MNIST_{GAN}$   
Classifier only trained on  $MNIST_{GAN}$

$5 \times C-MNIST_{original+GAN}$   
Classifier only trained on  $MNIST_{original+GAN}$

$5 \times C-MNIST_{original+font}$   
Classifier only trained on  $MNIST_{original+font}$

Each classifier was tested on the MNIST test set which resulted in an accuracy percentage.

## 4. Results

To validate whether the GAN was creating useful images, we looked at the resulting images in  $\text{MNIST}_{\text{GAN}}$ :

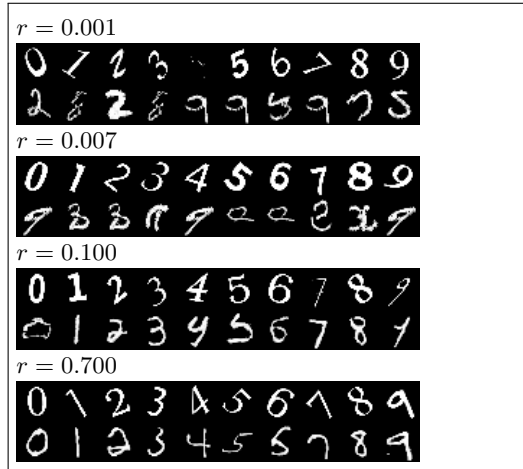


Figure 3. Example images of the MNIST font dataset. Top of each image is from  $\text{MNIST}_{\text{font}}$ , bottom is the generated image in  $\text{MNIST}_{\text{GAN}}$

From visual inspection it is apparent that the GAN has trouble keeping the labels consistent if there is not enough real training data. An example of this are the 1 and the 2 of  $r = 0.007$  in figure 3: they both result in something that looks like the same 8 instead of something that looks like a 1 or 2 respectively. This means that the classifier trained on this new dataset will receive two very similar samples with conflicting labels. We measured the accuracy on MNIST of  $\text{C-MNIST}_{\text{GAN}}$  and found low accuracy for lower ratios. This supports the conclusion that there is an issue with labeling.

We measured the performance of  $\text{C-MNIST}_{\text{original+GAN}}$  versus  $\text{C-MNIST}_{\text{original}}$ . It showed that even though the generated samples are not good samples for their target class,  $\text{C-MNIST}_{\text{original+GAN}}$  is much more accurate than  $\text{C-MNIST}_{\text{original}}$ . This effect can partially be explained by results shown by Sukhbaatar et al. that indicate that deep learning is robust to massive label noise. [10] This would mean that the network is still capable of learning higher level features from the mislabeled samples and is able to successfully ignore the bad labeling. This indicates that this technique is successfully able to close the distribution gap once a lower bound on the amount of available real images has been reached.

### 4.1. $\text{C-MNIST}_{\text{font}}$ performance

To make sure we actually test whether the GAN is able to close the distribution gap we looked at the accuracy of  $\text{C-MNIST}_{\text{font}}$ . For this dataset the error shows an inverse relation between the amount of samples and the performance

of the classifier as can be seen in figure 4. This indicates that a gap exists between the raw font dataset and the MNIST test dataset.

### 4.2. Classifier performance

We compared the accuracy of  $\text{C-MNIST}_{\text{font}}$ ,  $\text{C-MNIST}_{\text{GAN}}$ ,  $\text{C-MNIST}_{\text{original}}$  and  $\text{C-MNIST}_{\text{original+GAN}}$  to see whether training with the  $\text{MNIST}_{\text{original+GAN}}$  dataset is better than training with one of the individual datasets. The result of this comparison can be seen in figure 5.

There seems to be a minimum amount of real samples after which the GAN starts to produce meaningful data. In this case  $\text{C-MNIST}_{\text{original+GAN}}$  is only more accurate than the  $\text{C-MNIST}_{\text{original}}$  when the GAN is trained on more than 385 real images ( $r = 0.007$ ).

Using MNIST as a test case, this technique is able to close the distribution gap even with as little 0.7% real data in the resulting dataset.

### 4.3. $\text{C-MNIST}_{\text{original}}$ against $\text{C-MNIST}_{\text{original+font}}$

To confirm that the improved results of  $\text{C-MNIST}_{\text{original+GAN}}$  can not be solely explained by the addition of more data, we compared  $\text{C-MNIST}_{\text{original}}$  with  $\text{C-MNIST}_{\text{original+font}}$  to see if the addition of font data increased the accuracy. From the results in figure 6 we had to conclude that  $\text{C-MNIST}_{\text{original+font}}$  has performance roughly equal to  $\text{C-MNIST}_{\text{original}}$  and therefore cannot explain the additional accuracy of  $\text{C-MNIST}_{\text{original+GAN}}$ .

## 5. Discussion

Due to a limited amount of time, no optimization has been done on the hyperparameters of the GAN. Optimizing these hyperparameters might further improve classification accuracy which should result in a further reduction in the amount of required samples from the real dataset.

## 6. Conclusion

Using MNIST as an example, we were able to show that it's possible to create a synthetic dataset. We've shown that a distribution gap exists between the dataset we created and the real MNIST dataset and were able to close this gap using our described technique.

We've shown that GANs can be used to inflate trainings by reducing the gap between synthetic and real datasets. Furthermore, they can do so with very little real training data.

## References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané,

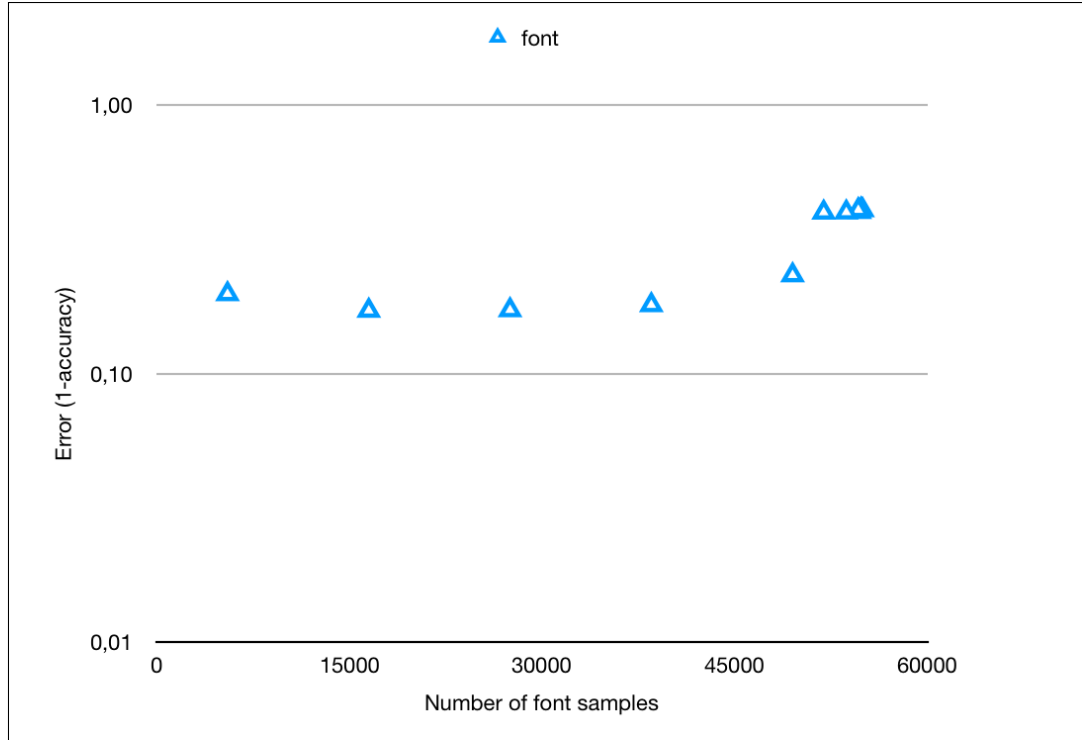


Figure 4. Performance of C-MNIST<sub>font</sub> with varying amounts of MNIST<sub>font</sub> training data. Performance decreases with more training samples indicating that MNIST<sub>original</sub> and MNIST<sub>font</sub> have different characteristics

- R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, July 2017.
- [3] J. Cano-Perez, J.-C. Pérez-Cortes, J. Arl, and R. Llobet. Training set expansion in handwritten character recognition. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops SSPR 2002 and SPR 2002 Windsor, Ontario, Canada, August 6–9, 2002 Proceedings*, 01 2003.
- [4] L. Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, Nov 2012.
- [5] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. *Domain-Adversarial Training of Neural Networks*, pages 189–209. Springer International Publishing, Cham, 2017.
- [6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press.
- [7] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Understanding realworld indoor scenes with synthetic data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4077–4085, June 2016.
- [8] H. S. B. Jean Nonnemaker. Using synthetic data safely in classification. In *PROCEEDINGS VOLUME 7247 IS&T/SPIE ELECTRONIC IMAGING — 18-22 JANUARY 2009*, volume 7247, pages 7247 – 7247 – 11, 01 2009.
- [9] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1857–1865, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.
- [10] S. Sukhbaatar, J. Bruna, M. Paluri, L. D. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. In *ICLR Workshop track*, 2014.

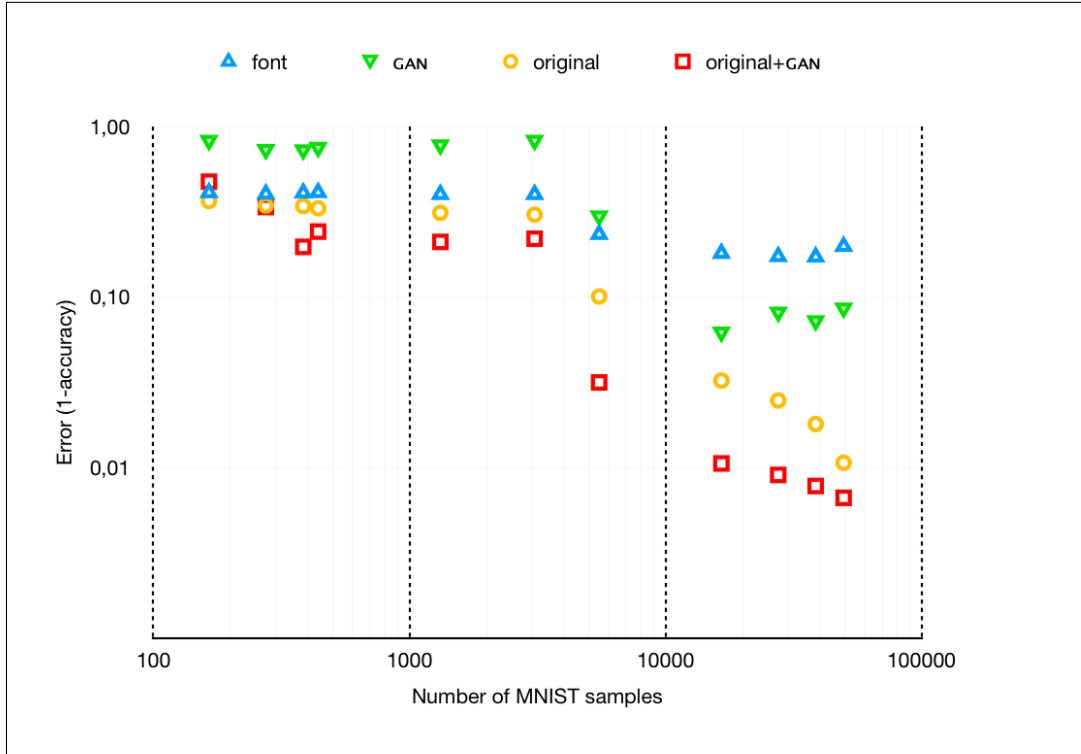


Figure 5. Performance of classifiers with different amounts of MNIST training data. Training with the GAN data improves results after a minimal amount of original MNIST samples

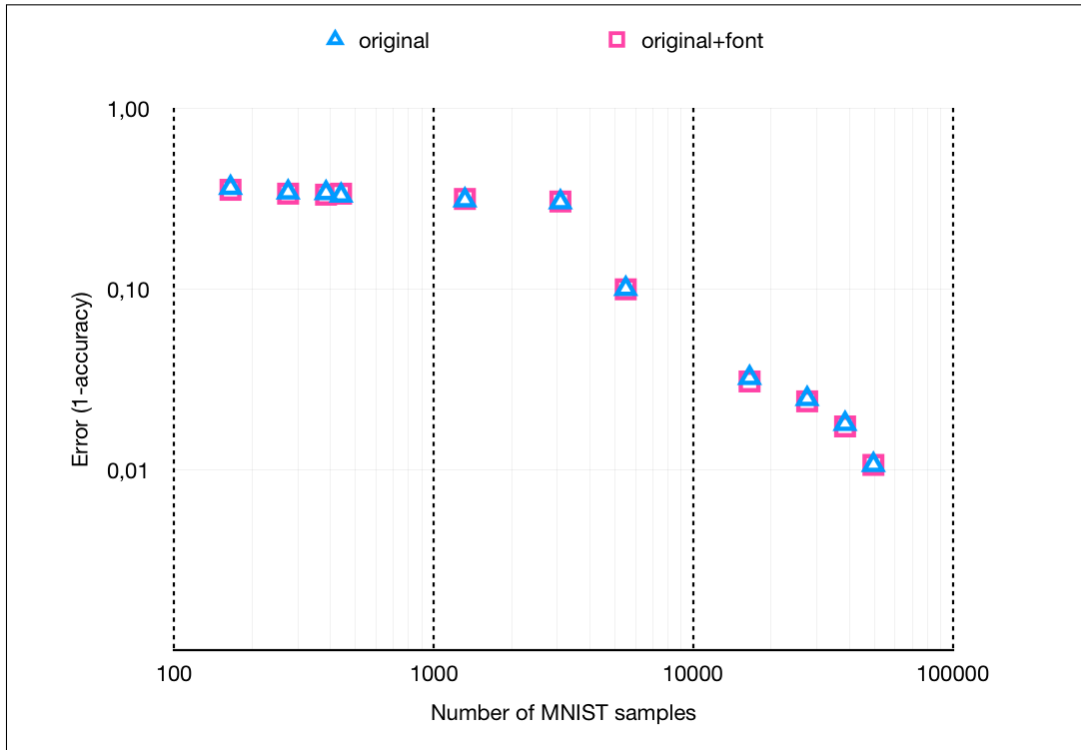


Figure 6. Performance of C-MNIST<sub>original+font</sub> versus the performance of C-MNIST<sub>original</sub> for varying amounts of training data. Performance is almost the same indicating no advantage to including font data in the training set directly.