A method to improve classifier performance using Generative Adversarial Networks(GANs) based data augmentation on the Kannada MNIST dataset

prateik@stanford.edu

Motivation

Deep Learning techniques are typically data intensive systems that demand a large amount of training data to be used to obtain acceptable real world performance. For certain classes of tasks like handwritten digits recognition, as long as the digits are identifiable by a human it does not matter how the data is generated. Therefore, exploring the idea of using data augmentation and GANs for developing synthetic training data is interesting.

Data and Preprocessing

Kannada is a regional language spoken in South India by over 44 million people. The Kannada MNIST dataset has been recently (Aug 2019).

೦	೧	೨	೩	೪	೫	೬	೭	೮	೯
0	1	2	3	4	5	6	7	8	9

The input to the algorithm is a gray scale image which is 28 by 28 pixels in dimensions. A Convolutional Neural Network (CNN) is used to output a predicted digit out of 10 classes.

Keras based ImageDataGenerator was used for Data Augmentation with the following specifications:

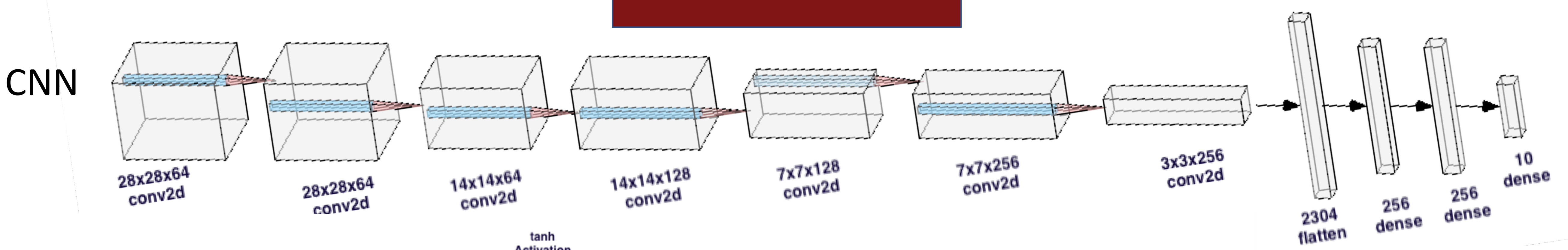
Parameter	Value
rescale	1/255
rotation range	10
width shift range	0.25
height shift range	0.25
shear range	0.1
zoom range	0.25
horizontal flip	False

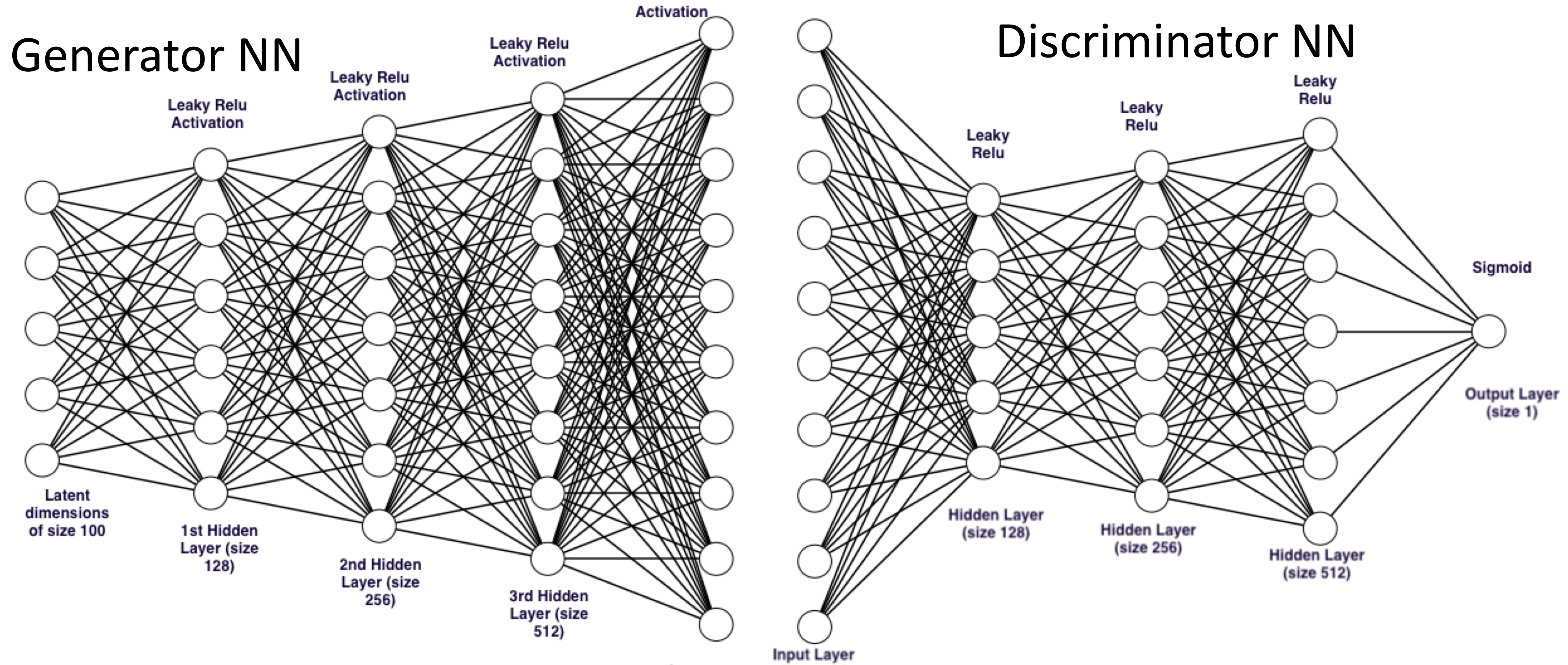
Dataset Size Spec

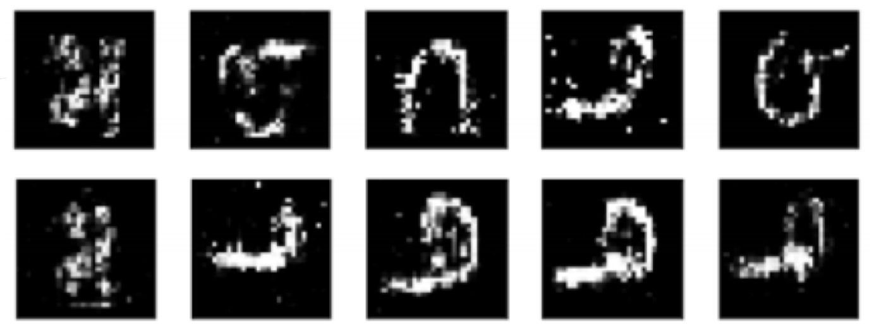
Dataset	Size	
	cnn	cnn + gan
Training	54K	53.9K + 0.1 K
Validation	6K	6K
Test	10K	10K

Loss function $L = -\Sigma_{c=1}^{M} y_{i,c} \log(p_{i,c})$

Models







- **Data :** Rescale the MNIST images to be between -1 and 1.
- **Generator** : Simple fully connected neural network, Leaky ReLU activation and BatchNormalization.
- **Discriminator** : Simple fully connected neural network and Leaky ReLU activation. The last activation is sigmoid.
- **Loss :** binary_crossentropy
- **Optimizer :** Adam(lr=0.0002, beta_1=0.5)
- **Batch_size :** 64
- **Epochs** : 100

Hyperparameter Tuning

- **Batch Size Tuning**

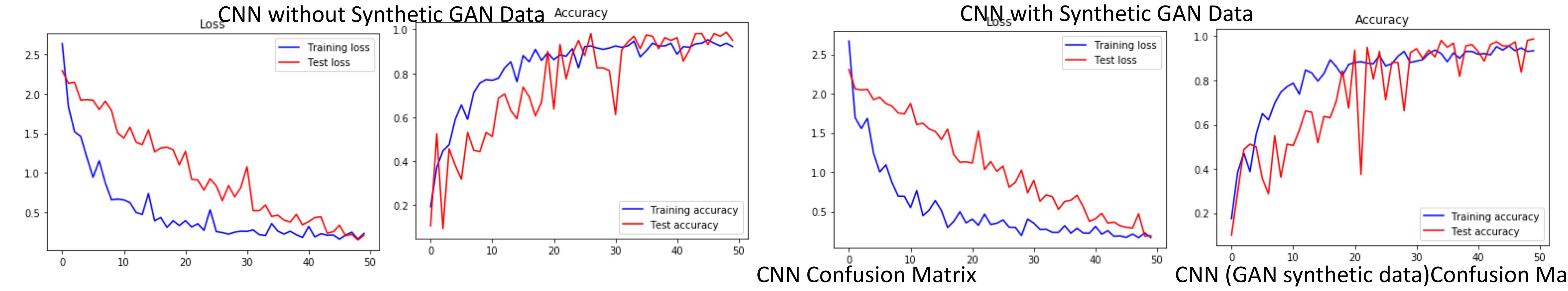
Batch Size	Test Accuracy(%)	Test Accuracy(%with GAN)
64	52.28	70.07
512	66.15	72.8
1024	75.37	68.89
2048	78.89	68.07

- **Learning Rates used :** 0.02 and 0.002
- **Learning Rate Decay** : Learning rate * 0.99 ^ epoch
- **Batch Normalization**: momentum = 0.1
- **Hyperparameters used for the Analysis:**

Hyperparameter	Value
learning rate	0.002
mini-batch size	64
epochs	50
Optimizer	RMSProp

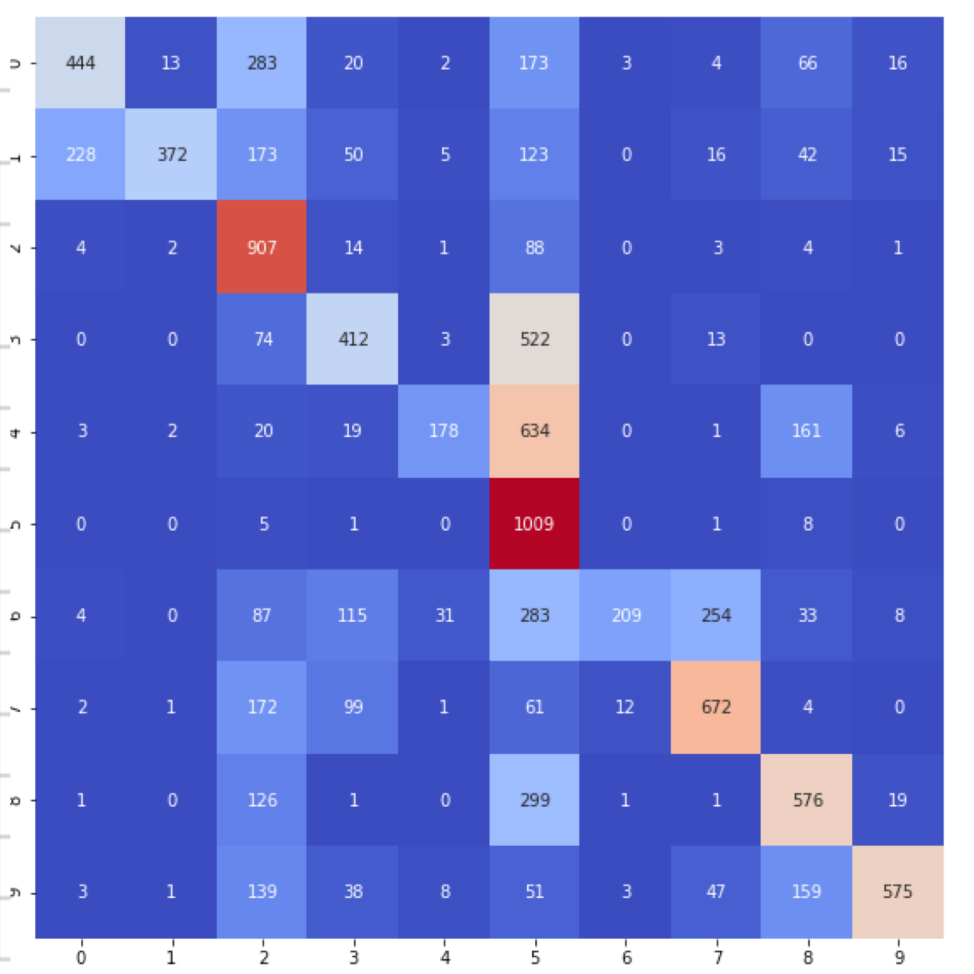
Results and Discussions

X Axis Loss/Accuracy Value
Y axis number of epochs



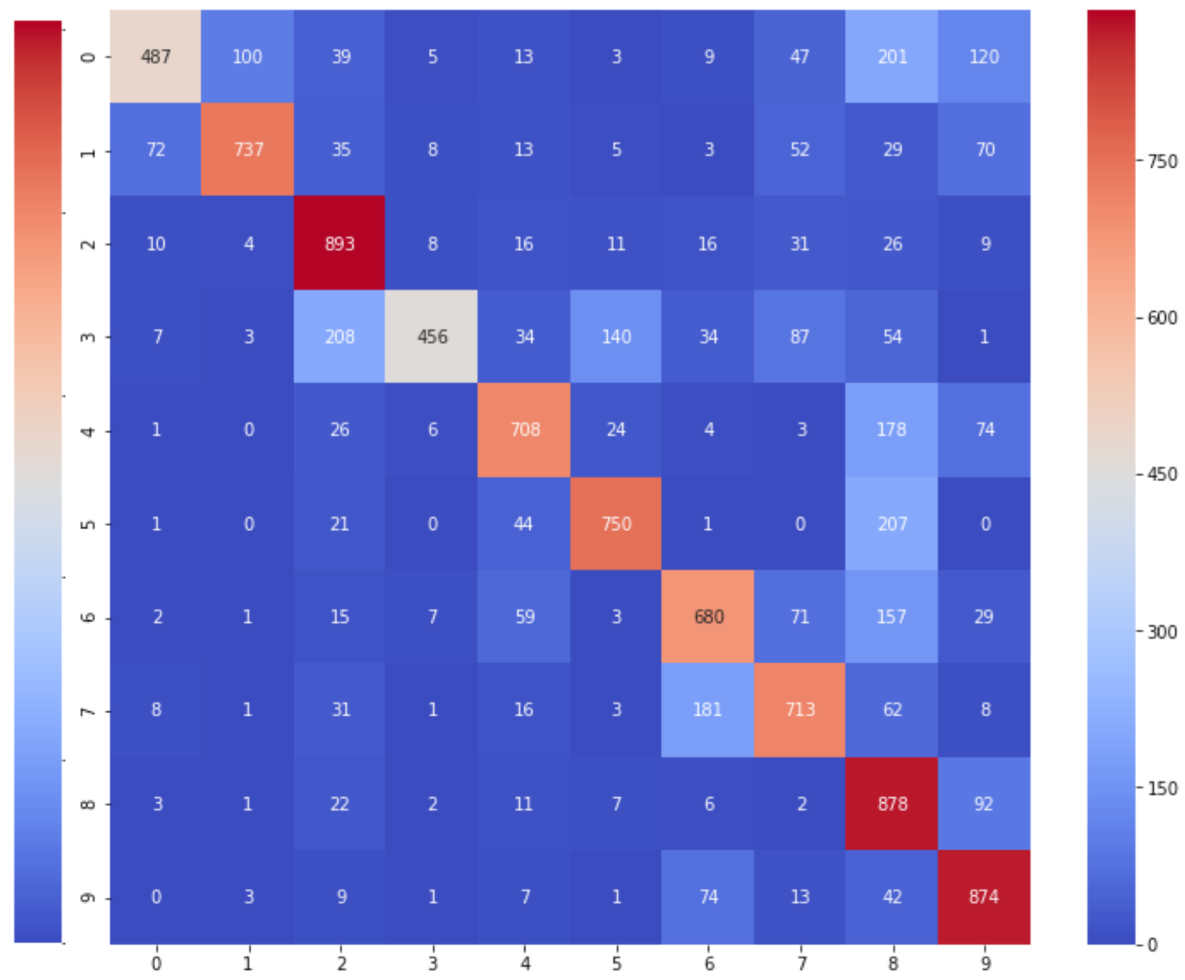
Class	Precision		recall		fl-score		support
	cnn	cnn + gan	cnn	cnn + gan	cnn	cnn + gan	
class 0	0.64	0.82	0.43	0.48	0.52	0.6	1024
class 1	0.95	0.87	0.36	0.72	0.53	0.79	1024
class 2	0.46	0.69	0.89	0.87	0.6	0.77	1024
class 3	0.54	0.92	0.4	0.45	0.46	0.6	1024
class 4	0.78	0.77	0.17	0.69	0.28	0.73	1024
class 5	0.31	0.79	0.99	0.73	0.47	0.76	1024
class 6	0.92	0.67	0.2	0.66	0.33	0.67	1024
class 7	0.66	0.7	0.66	0.7	0.66	0.7	1024
class 8	0.55	0.48	0.56	0.86	0.55	0.61	1024
class 9	0.9	0.68	0.56	0.85	0.69	0.76	1024
micro avg	0.52	0.7	0.52	0.7	0.52	0.7	10240
macro avg	0.67	0.74	0.52	0.7	0.51	0.7	10240
weighted avg	0.67	0.74	0.52	0.7	0.51	0.7	10240

CNN Confusion Matrix



	0	1	2	3	4	5	6	7	8	9
0	444	13	283	20	2	173	3	4	66	16
1	228	372	173	50	5	123	0	16	42	15
2	4	2	907	14	1	88	0	3	4	1
3	0	0	74	412	3	522	0	13	0	0
4	3	2	20	19	178	634	0	1	161	6
5	0	0	5	1	0	1009	0	1	8	0
6	4	0	87	115	31	283	209	254	33	8
7	2	1	172	99	1	61	12	672	4	0
8	1	0	126	1	0	299	1	1	576	19
9	3	1	139	38	8	51	3	47	159	575

CNN (GAN synthetic data)Confusion Matrix



	0	1	2	3	4	5	6	7	8	9
0	487	100	39	5	13	3	9	47	201	120
1	72	737	35	8	13	5	3	52	29	70
2	10	4	893	8	16	11	16	31	26	9
3	7	3	208	456	34	140	34	87	54	1
4	1	0	26	6	708	24	4	3	178	74
5	1	0	21	0	44	750	1	0	207	0
6	2	1	15	7	59	3	680	71	157	29
7	8	1	31	1	16	3	181	713	62	8
8	3	1	22	2	11	7	6	2	878	92
9	0	3	9	1	7	1	74	13	42	874

Conclusion

In this project, a data augmentation method using Generative Adversarial Networks(GANs) to improve the performance of a Convolution Neural Network(CNN) for the task of classification of Kannada MNIST digits is presented. This technique replaced 100 images from the original dataset with synthetic generated data. The test accuracy is shown to improve from 52.28 % to 70.07 % and the CNN F1 score(macro avg.) has improved from 0.51 to 0.70.

References

- Kannada-MNIST: A new handwritten digits dataset for the Kannada language, Prabhu, Vinay Uday, arXiv preprint arXiv:1908.01242, 2019
- Web Reference: https://www.kaggle.com/bustam/cnn-in-keras-for-kannada-digits
- Web Reference:https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-an-mnist-handwritten-digits-from-scratch-in-keras/