

3rd International Conference on Computer Science and Computational Intelligence 2018

# Deep Learning for Imbalance Data Classification using Class Expert Generative Adversarial Network

Fanny<sup>a,\*</sup>, Tjeng Wawan Cenggoro<sup>a,b</sup>

<sup>a</sup>Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480

<sup>b</sup>Bioinformatics and Data Science Research Center, Bina Nusantara University, Jakarta, Indonesia 11480

---

## Abstract

Without any specific way for imbalance data classification, artificial intelligence algorithm cannot recognize data from minority classes easily. In general, modifying the existing algorithm by assuming that the training data is imbalanced, is the only way to handle imbalance data. However, for a normal data handling, this way mostly produces a deficient result. In this research, we propose a class expert generative adversarial network (CE-GAN) as the solution for imbalance data classification. CE-GAN is a modification in deep learning algorithm architecture that does not have an assumption that the training data is imbalance data. Moreover, CE-GAN is designed to identify more detail about the character of each class before classification step. CE-GAN has been proved in this research to give a good performance for imbalance data classification.

© 2018 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Selection and peer-review under responsibility of the 3rd International Conference on Computer Science and Computational Intelligence 2018.

**Keywords:** imbalance data; classification; deep learning; generative adversarial network

---

## 1. Introduction

Classification of imbalance data is one of a classic problem in the artificial intelligence area, especially for classification in machine learning. Imbalance data has been proved can decrease the performance of machine learning algorithm<sup>1</sup>, where imbalance data means the total of data from each class is significantly different. The example of imbalance data can be seen in the works from Cenggoro et al.<sup>2,3</sup>. The data from those researches shows that classes of an urban area is up to 62.45% from the total area, while the class of an open area is only 0.02% from the total area. Usually, machine learning algorithm cannot learn this kind of data, especially for minority classes because the information is covered by majority classes data. There are some researches in deep learning area that is trying to prevent classification in imbalance data<sup>4,5</sup>. However, these researches still use algorithm that assumes the training dataset is imbalance. This assumption can decrease the performance of the algorithm if it is used for balance training dataset.

---

\* Corresponding author. Tel.: +62-21-543-5830 ext 2316 ; fax: +62-21-530-1668.

E-mail address: [fanny.sa@binus.edu](mailto:fanny.sa@binus.edu)

## 2. Related Works

Several works have been done to solve imbalance data problem in deep learning. Huang et al.<sup>4</sup> handled imbalance data by integrating sampling process into Convolutional Neural Network (CNN)<sup>6</sup>. The sampling process was done by dividing the data into several clusters, then a pair of data based on the determined condition was retrieved. These pairs of data were then inputted to the CNN. After this, the process was then repeated again from clustering until the model reaches the convergences. Yan et al.<sup>7</sup> and Khan et al.<sup>5</sup> handled imbalance data by modifying a CNN technique. Yan et al. modified it by combining bootstrapping and CNN while Khan et al. designed a cost function that is tailored to CNN characteristics to solve the same problem. In addition, there are several researches that explained how to handle imbalance data, even though their main focus was not specifically to deal with imbalance data. Wei et al.<sup>8</sup> handled an imbalance data in object recognition by designing an image-cutting technique that is used as an additional training data. George<sup>9</sup> and Mostajabi et al.<sup>10</sup> used cost-sensitive to handle imbalance data in scene-parsing. The technique that was used in these researches is designed only for the dataset that used in each research. Thus, it would be difficult to implement the technique for different dataset.

## 3. Proposed Method

### 3.1. Class Experts

Class Experts (CE) is a concept proposed by Cenggoro to overcome imbalance problem in deep learning<sup>11</sup>. This concept utilizes a paradigm of arranging neural network layers in parallel, which each layer has been pretrained to recognize characteristic of a single class in the training data. These layers are pretrained with autoencoder-style algorithm. Cenggoro specifically use Variational Autoencoder in his work<sup>11</sup>. After the pretraining, the weights are transferred to main neural network model and trained supervised like a normal classification task. Fig. 1 depicted the main model with parallel layers grouped as Class Experts Layer, which each module in this layer is pretrained using single-class dataset.

### 3.2. Class Experts Generative Adversarial Network

Different from the work of Cenggoro<sup>11</sup>, we propose the use of Generative Adversarial Network (GAN)<sup>12</sup> for pretraining algorithm instead of autoencoder-style algorithm. In contrast to autoencoder-style algorithm, GAN has more theoretical benefit to encode single-class characteristics useful for classification task. The most noticeable benefit of GAN for classification task is that it incorporates the use of discriminative model in its process. When trained with only single class, the GAN discriminative model can be viewed as a classifier that is able to tell whether input data is from the assigned class or not. In other words, this discriminative model encode distribution  $p(y_i|x)$ , where  $y$  is the decision whether the input data  $x$  is from class  $i$  or not. This is contrast to autoencoder-style algorithm that encodes distribution  $p(x)$ . Therefore, the weights of GAN discriminative model are more natural to be transferred to the main classifier model, which optimize distribution  $p(y|x)$ , where  $y$  is the label to determine the class of input data  $x$  in multi-class setting. Fig. 2 illustrates the process of GAN pretrained weights transfer to main CNN model in the proposed algorithm. After the weight transfer, we can choose whether to keep the weight frozen or not during the finetuning phase. To the rest of this paper, we will call the proposed algorithm with frozen weights as frozen class expert generative adversarial network (FCE-GAN). For the counterpart with trainable weights, it will be called as class expert generative adversarial network (CE-GAN).

## 4. Experiment Setting

### 4.1. Data Preparation

The data used in this research is sliced from CelebA dataset<sup>13</sup>. Specifically, only first five classes of CelebA dataset used in this research: 5 O'Clock Shadow, Arched Eyebrows, Attractive, Bags Under Eyes, ands Bald. The limitation of dataset used is necessary for the conciseness of balancing effect analysis in this research. The whole sliced dataset

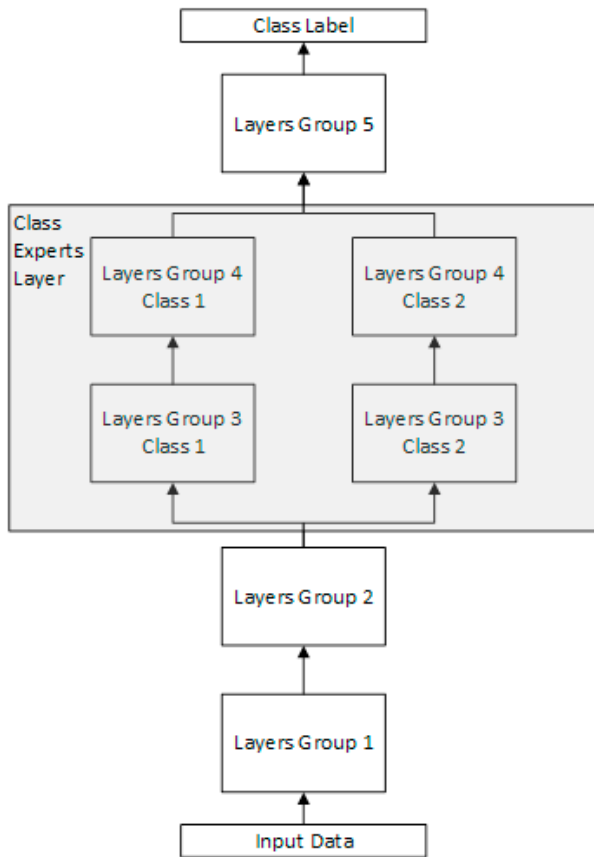


Fig. 1: Example of Main Neural Network Model with Class Experts Layer

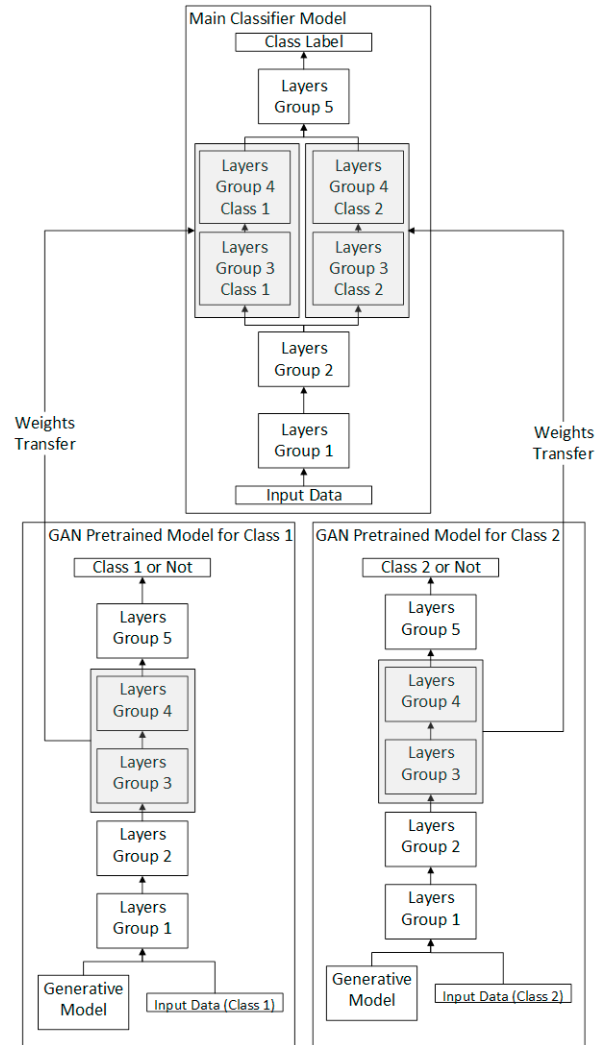


Fig. 2: Illustration of GAN Pretrained Weights Transfer Process for Two Classes

is then divided into training, validation, and testing sub-dataset, with the ratio of 6:2:2. The distribution of training sub-dataset is visualized in Fig. 3. From this figure, we can see that the training dataset is imbalance, with Attractive as the majority class and Bald as the most minority class.

#### 4.2. Models in Comparison

In this research, performance of four models are compared: two proposed algorithms (CE-GAN CNN and FCE-GAN CNN) and two other algorithms from previous researches. The other two compared algorithm are Bootstrapping CNN<sup>7</sup> and CosenCNN<sup>5</sup>, which are CNN models designed to tackle imbalance problems. In addition, we also display the result from pure baseline CNN, which is explained in sub-chapter 4.3.

#### 4.3. Model Configuration

The CNN used in all compared algorithms are based on a same CNN architecture, which is called as baseline CNN architecture in this paper. The objective of using this baseline CNN is to guarantee that the performance comparison is

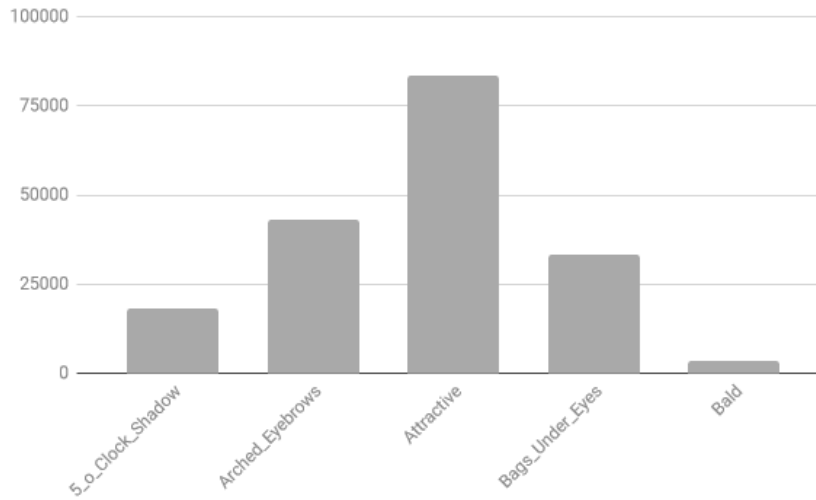


Fig. 3: Classes Distribution of Training Dataset

not affected by the different CNN capability used by each compared algorithm. The architecture used for the baseline CNN is taken from the discriminative CNN used in the GAN model, which is described in table 2. Layer 1 to 6 of the discriminative architecture is based on the reverse of layer 1 to 6 of the generative CNN architecture employed in the GAN model with a little adjustment so that discriminative CNN has same kernel size for all its convolution layer. The generative CNN architecture is described in table 1. The architecture of generative CNN is designed specifically to generate image with the size of 218x178, same as the original images size, from a randomly generated input vector with size of 1x100.

For a better performance, we employ several batch normalizations within the discriminative CNN architecture, noted as BN in table 1 and 2. Batch normalization has been proved to increase deep learning model performance<sup>14</sup>. We also use rectified linear unit (ReLU) activation functions for our CNN. ReLU has been proved to be able to boost the performance of CNN with deep layers<sup>15</sup>.

The optimization method used for all compared models is Adam<sup>16</sup>, which is already proven to be able to optimize deep learning model faster. The learning rate used is 0.001, as suggested by Kingma & Ba<sup>16</sup>. To determine the number of consecutive layers used as class experts in the proposed models, a pre-experiment is run to see the validation accuracy of different number of layer in CE for CE GAN CNN in 5<sup>th</sup> iteration. The result of this pre-experiment is tabularized in table 3. From this table, layer 3, 4, 5, and 6 give the best results when used in CE of the CE GAN CNN. Therefore, this CE architecture configuration is used for performance comparison of proposed models with other algorithms.

The optimization method used for all compared models is Adam<sup>16</sup>, which is already proven to be able to optimize deep learning model faster. The learning rate used is 0.001, as suggested by Kingma & Ba<sup>16</sup>. To determine the number of consecutive layers used as class experts in the proposed models, a pre-experiment is run to see the validation accuracy of different number of layer in CE for CE-GAN CNN in 5th iteration. The result of this pre-experiment is tabularized in table 3. From this table, layer 3, 4, 5, and 6 give the best results when used in CE of the CE-GAN CNN. Therefore, this CE architecture configuration is used for performance comparison of proposed models with other algorithms.

Table 1: Generator CNN Architecture

#Layer	Layer Type	Layer Description
1	Deconvolution+BN+ReLU	Size = 5x4x192 Stride = 1 Padding Zero = No Input Size = 1x100x1 Output Size = 5x4x192
2	Deconvolution+BN+ReLU	Size = 4x4x192 Stride = 5 Padding Zero = Yes Output Size = 25x20x192
3	Deconvolution+BN+ReLU	Size = 3x3x192 Stride = 1 Padding Zero = No Output Size = 27x22x192
4	Deconvolution+BN+ReLU	Size = 4x4x96 Stride = 2 Padding Zero = Yes Output Size = 54x44x96
5	Deconvolution+BN+ReLU	Size = 4x4x48 Stride = 2 Padding Zero = Yes Output Size = 108x88x48
6	Deconvolution+BN+ReLU	Size = 4x4x24 Stride = 2 Padding Zero = No Output Size = 218x178x24
7	Deconvolution	Size = 4x4x3 Stride = 1 Padding Zero = Yes Output Size = 218x178x3

Table 2: Discriminator CNN Architecture

#Layer	Layer Type	Layer Description
1	Convolution+ReLU	Size = 4x4x24 Stride = 2 Padding Zero = Yes Input Size = 218x178x3 Output Size = 109x89x24
2	Convolution+BN+ReLU	Size = 4x4x48 Stride = 2 Padding Zero = Yes Output Size = 55x45x48
3	Convolution+BN+ReLU	Size = 4x4x96 Stride = 2 Padding Zero = Yes Output Size = 28x23x96
4	Convolution+BN+ReLU	Size = 4x4x192 Stride = 2 Padding Zero = Yes Output Size = 14x12x192
5	Convolution+BN+ReLU	Size = 4x4x192 Stride = 2 Padding Zero = Yes Output Size = 7x6x192
6	Convolution+BN+ReLU	Size = 4x4x24 Stride = 2 Padding Zero = Yes Output Size = 4x3x192
7	Convolution	Size = 4x4x3 Stride = 1 Padding Zero = No Output Size = 3x2x1
8	Fully Connected + ReLU	#Neuron = 200 Output Size = 200x1
9	Fully Connected + Sigmoid	#Neuron = 5 Output Size = 5x1

Table 3: Pre-Experiment Result on Determining Number of Layers Used as CE.

Layers Used as CE	Validation Accuracy
6	75.59 %
5 and 6	75.59 %
4, 5, and 6	74.92 %
<b>3, 4, 5, and 6</b>	<b>76.27 %</b>
2, 3, 4, 5, and 6	73.22 %

#### 4.4. Analyzing Results

The metrics used to assess performance of all compared algorithms are classification accuracy and precision. Classification accuracy used to assess overall algorithm performance. Equation 1 shows the calculation of classification accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Where:

- TP = True Positive, the number of data that are correctly classified as a particular class
- TN = True Negative, the number of data that are correctly not classified as a particular class
- FP = False Positive, the number of data that are mistakenly classified as a particular class
- FN = False Negative, the number of data that are mistakenly not classified as a particular class

To assess the balancing effect of all compared algorithms, we consider the use of classification precision metric. Classification precision can be calculated using equation 2. By using precision, we can see the performance of each algorithm in terms of correctly classify minority class (TP), without being shadowed by the sheer number of negative examples from minority class.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

### 5. Experiment Results and Analysis

The performance of all GAN pretrained models can be seen by looking at the sample of generated images by the model in figure 4. We can see that the models generally able to catch unique features for each classes. For instance, all generated images in class 5 o'clock shadow, arched eyebrows, bags under eyes, and bald indeed has 5 o'clock shadow, arched eyebrow, bags under eyes, and baldness. While it would be difficult to see if model for subjective class such as attractive is able to catch desirable feature, we can still see that the model is able to generate reasonable face. It is important to make sure that the model can catch unique features of each class, as the idea of CE is to transfer these features to the classification model.

Table 4 shows classification accuracy performance of all compared algorithms. We can see that the best overall accuracy is achieved by CE-GAN CNN. The best classification accuracy of each class also performed by CE-GAN CNN, with the only exception for class Bags Under Eyes, which only 0.45% less than the best accuracy achieved by baseline CNN.

Table 4: Compared Algorithms Classification Accuracy Performance.

Algorithm	Per-Class Accuracy					Overall Accuracy
	5 O'Clock Shadow	Arched Eyebrows	Attractive	Bags Under Eyes	Bald	
CNN with Bootstrapping	84.85	67.14	71.26	73.80	95.97	78.61
CosenCNN	86.43	67.95	72.60	75.39	97.13	79.90
Baseline CNN	92.04	78.95	81.23	<b>85.03</b>	98.40	87.13
CE-GAN CNN	<b>92.71</b>	<b>78.98</b>	<b>82.09</b>	84.58	<b>98.56</b>	<b>87.38</b>
FCE-GAN CNN	88.35	68.86	78.73	81.52	97.48	82.99

As previously discussed, it is necessary to see the performance of all compared algorithms using classification precision metric to analyze their balancing effect. These precision results are tabularized in table 5. We can see from this table that CE-GAN CNN has the best balancing effect among all compared algorithms. This can be seen from the fact that only CE-GAN CNN can increase all the precision of minority classes from baseline CNN. Not only that, CE-GAN CNN also able to increase the most majority class precision by 1.57%. Meanwhile, the other approach proposed



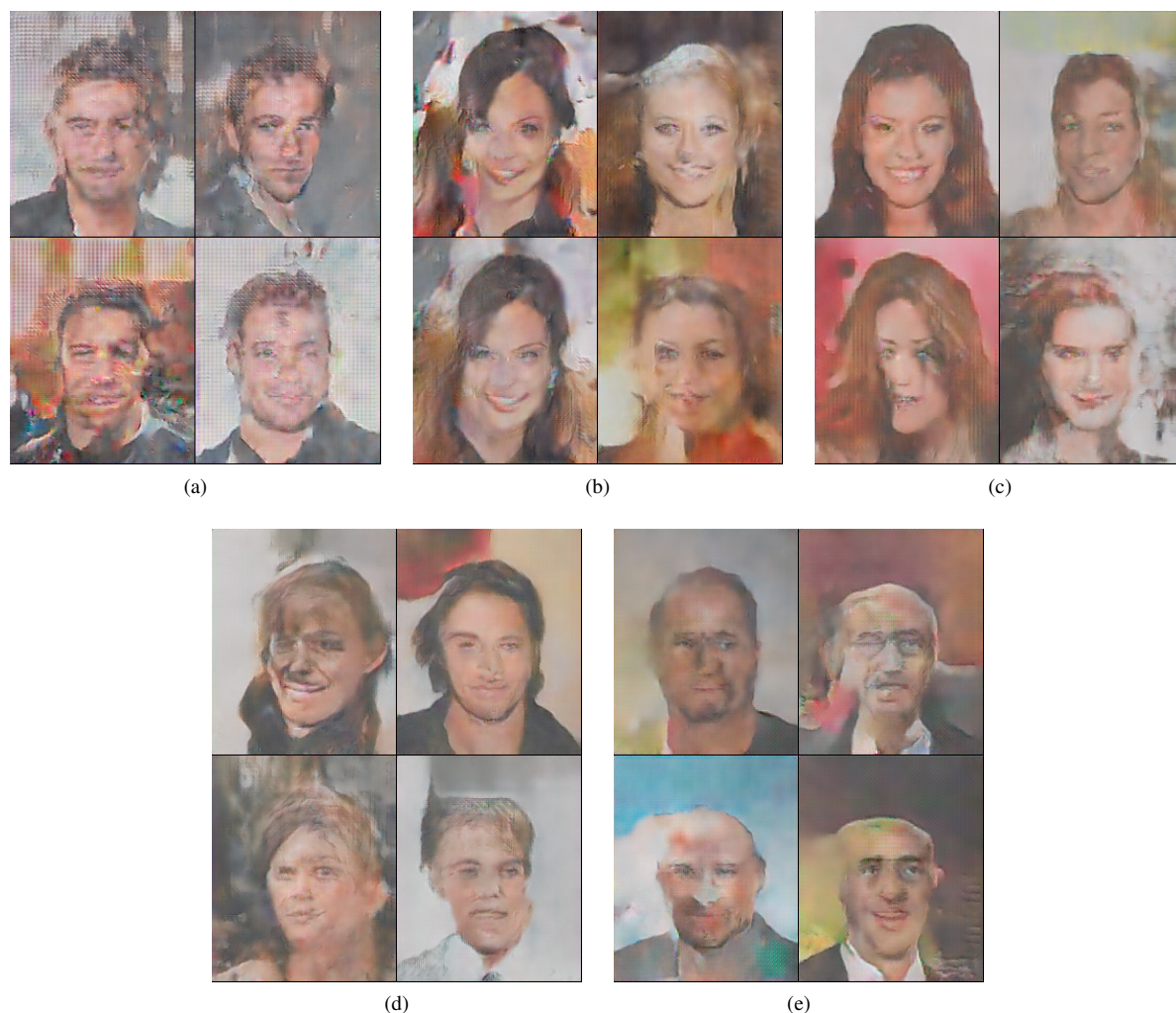


Fig. 4: Sample of GAN Generated Images Pretrained with class: (a) 5 O'Clock Shadow; (b) Arched Eyebrows; (c) Attractive; (d) Bags Under Eyes; (e) Bald.

in this paper, FCE-GAN, is not performing as expected. While it only differs from CE-GAN CNN in keeping the transferred weights fixed, the precision performance of FCE-GAN CNN is significantly lower than CE-GAN CNN. Its balancing effect even lower than CNN with bootstrapping, which can increase two of four minority classes precision.

Table 5: Compared Algorithms Classification Precision Performance.

Algorithm	Per-Class Accuracy					Overall Precision
	5 O'Clock Shadow	Arched Eyebrows	Attractive	Bags Under Eyes	Bald	
CNN with Bootstrapping	<b>82.30</b>	38.89	71.05	8.53	69.74	52.54
CosenCNN	9.18	48.52	<b>89.81</b>	26.55	0.00	58.54
BaseLine CNN	50.10	73.44	84.08	73.40	69.50	76.02
CE-GAN CNN	65.20	<b>76.00</b>	85.65	<b>77.35</b>	<b>77.78</b>	<b>79.64</b>
FCE-GAN CNN	56.82	35.63	86.62	53.45	65.72	64.30

## 6. Conclusion

In this paper, we demonstrate the benefits of using CE paradigm and GAN pretraining in countering imbalance data problem in CNN. The utilization of CE is able not only to give a better balancing precision, but also to increase the overall accuracy of standard CNN. The balancing effect of given by CE is also better than other compared algorithm in this paper. However, to achieve these benefits, the transferred weights from pretrained GAN model need to be allowed to change as the main CNN model learn supervisedly.

We might notice that the type of GAN employed in this research is the original GAN. There has been several type of GANs that modify the original GAN and successfully improve its performance. Therefore, it would be interesting to see the effect of using CE concept with other type of GANs.

## Acknowledgements

This research is funded by Bina Nusantara University Applied Technology Grant 2017. We would also like to acknowledge NVIDIA Bina Nusantara University AI R&D Center for the GPU used in this research.

## References

1. Dalay, A., Shanahan, M., Kelly, J.. Tackling Class Imbalance with Deep Convolutional Neural Networks; 2014.
2. Cenggoro, T.W., Isa, S.M., Kusuma, G.P.. Construction of Jakarta Land Use/Land Cover dataset using classification method. In: *2016 IEEE Region 10 Symposium (TENSYP)*. IEEE. ISBN 978-1-5090-0931-2; 2016, p. 337–342. doi:\bibinfo{doi}{10.1109/TENCONSpring.2016.7519429}.
3. Cenggoro, T.W., Isa, S.M., Kusuma, G.P., Pardamean, B.. Classification of imbalanced land-use/land-cover data using variational semi-supervised learning. In: *2017 International Conference on Innovative and Creative Information Technology (ICITech)*. IEEE. ISBN 978-1-5386-4046-3; 2017, p. 1–6. doi:\bibinfo{doi}{10.1109/INNOCIT.2017.8319149}.
4. Huang, C., Li, Y., Loy, C.C., Tang, X.. Learning Deep Representation for Imbalanced Classification. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. ISBN 978-1-4673-8851-1; 2016, p. 5375–5384. doi:\bibinfo{doi}{10.1109/CVPR.2016.580}.
5. Khan, S.H., Hayat, M., Bennamoun, M., Soheli, F.A., Togneri, R.. Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data. *IEEE Transactions on Neural Networks and Learning Systems* 2017;:1–15doi:\bibinfo{doi}{10.1109/TNNLS.2017.2732482}. **1508.03422**.
6. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., et al. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* 1989;1(4):541–551. doi:\bibinfo{doi}{10.1162/neco.1989.1.4.541}.
7. Yan, Y., Chen, M., Shyu, M.I., Chen, S.c.. Deep Learning for Imbalanced Multimedia Data Classification. In: *2015 IEEE International Symposium on Multimedia (ISM)*. IEEE. ISBN 978-1-5090-0379-2; 2015, p. 483–488. doi:\bibinfo{doi}{10.1109/ISM.2015.126}.
8. Wei, X.s., Gao, B.b., Wu, J.. Deep Spatial Pyramid Ensemble for Cultural Event Recognition. In: *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. IEEE. ISBN 978-1-4673-9711-7; 2015, p. 280–286. doi:\bibinfo{doi}{10.1109/ICCVW.2015.45}.
9. George, M.. Image parsing with a wide range of classes and scene-level context. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. ISBN 978-1-4673-6964-0; 2015, p. 3622–3630. doi:\bibinfo{doi}{10.1109/CVPR.2015.7298985}. **1510.07136**.
10. Mostajabi, M., Yadollahpour, P., Shakhnarovich, G.. Feedforward semantic segmentation with zoom-out features. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. ISBN 978-1-4673-6964-0; 2015, p. 3376–3385. doi:\bibinfo{doi}{10.1109/CVPR.2015.7298959}. **1412.0774**.
11. Cenggoro, T.W.. *Imbalanced Learning in Land Use/Land Cover Data of Urban Area Using Variational Semi Supervised Learning Embedded with Class Expert Layer*. Master's thesis; Bina Nusantara University; 2016.
12. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. Generative Adversarial Networks. *Advances in Neural Information Processing Systems* 27 2014;:2672–2680**1406.2661**.
13. Liu, Z., Luo, P., Wang, X., Tang, X.. Deep Learning Face Attributes in the Wild. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE. ISBN 978-1-4673-8391-2; 2015, p. 3730–3738. doi:\bibinfo{doi}{10.1109/ICCV.2015.425}.
14. Ioffe, S., Szegedy, C.. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of The 32nd International Conference on Machine Learning* 2015;**1502.03167**.
15. Krizhevsky, A., Hinton, G.E.. ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems*. 2012, .
16. Kingma, D.P., Ba, J.. Adam: A Method for Stochastic Optimization. *The International Conference on Learning Representations* 2015 2015;:1–15**1412.6980**.