Re-Balancing of Long-Tailed Datasets Using CGAN Generation

Meher Mankikar, Deep Patel, Jimmy Zhang
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh, PA 15213
mmankika@andrew.cmu.edu, dmpatel@andrew.cmu.edu, jimmyzha@andrew.cmu.edu

Abstract

In this paper, we will explore the use of a Conditional GAN (CGAN) model to generate new instances to augment long-tailed datasets. When training a Convolutional Network on a long-tailed dataset, it is often the case that the classes with more data are predicted more frequently, resulting in low accuracy, especially for the long-tailed classes. Current techniques to solve this issue include random undersampling and oversampling. However, undersampling loses useful information for learning better representations in images and oversampling can overfit to the long-tail classes in training data. We try to fix these issues by rebalancing the dataset using synthetic samples created by a CGAN. We worked with a long-tailed version of the Fashion MNIST dataset. This dataset was used to train the CGAN and generate new samples. We then augmented the original long-tailed dataset with these synthetic samples to train the CNN. We vary the architecture of our CGAN and compare to multiple baselines to determine what form of augmentation leads to the best results.

1 Background / Literature

For the midway report, we gathered baselines by training a convolutional network on the original unbalanced dataset, the dataset after balancing with subsampling, and the dataset after balancing with oversampling. We found that the model trained on the unbalanced dataset had relatively low performance as it was overfitting to the dataset. Random undersampling slightly improved performance but not by much because we were removing a lot of information from the dataset. Random oversampling produced the best loss over the epochs. Note that our baselines for the midway report were trained using the NeRF dataset. For the final project, we have pivoted to use the Fashion MNIST dataset. Therefore, we have retrained these baselines and the results that we show later in this paper will be reflective of that.

In this project, we made use of conditional GANs which are the same as regular general adversarial models except that we pass in label information so that the model can generate samples for a specific label.

2 Related Work

The paper "Improved Sampling Techniques for Learning an Imbalanced Dataset" by Lauron and Pabico [1] discusses methods for learning using data with long tail distributions (imbalanced datasets). Issues with these kinds of datasets arise because skewed distribution of data over several classes in the dataset can lead to prediction for minority classes being largely inaccurate. The paper builds on two sampling techniques that are used to solve the issue of imbalanced datasets: random undersampling (RU) and random oversampling (RO). RU randomly removes entries from all classes except that with the smallest number of instances. This is continued until the number of instances for each class is equal. On the other hand, RO adds data by randomly replicating instances already in the dataset of all the minority classes until the number of instances for each class are equal.

The survey paper "A survey on generative adversarial networks for imbalance problems in computer vision tasks"[3] discusses the duality of GAN-based oversampling techniques to counteract imbalanced datasets. The paper cites a few drawbacks of data-level balancing approaches that artificially inflate the dataset. The synthetically generated data may not be truly representative of the training set and may exaggerate the large intra-class variability and small inter-class variability of the data. Despite this, there have been many successes of GAN-based balancing, especially in medical image datasets. The survey paper cites experiments with statistically significant increases in baseline unbalanced models in the fields of industrial defect studies, breast cancer detection, and X-ray images.

3 Methods / Models

In order to improve on the results of random undersampling (RU) and random oversampling (RO), we first attempted to train a NeRF model in order to generate new synthetic samples to rebalance our dataset before training the convolutional network. While we were able to get the NeRF model to run, the training was extremely slow as it would have taken more than two days to train a sufficient model to generate new samples on a GPU. In addition, we would have had to train this NeRF model for each of the classes in our dataset separately. Furthermore, NeRF models require camera pose information and are trained specifically to scenes, making them less generalizabe to GANs. Given this large computational cost, we deemed this method infeasible and not as applicable for the timeframe that we have for this project.

Instead, we pivoted to creating a conditional GAN in order to generate new synthetic samples to balance our dataset. We expect that conditional GANs will give us the ability to create samples that capture the class-specific data distributions better than simply just copying instances of hte unbalanced dataset that we were given. There is a practical drawback to using GANs in which the predictions produced by the GAN will be noisy, and as a result, the images will not be perfectly sampled from a classes data distribution; however, we hope that the spread of the distribution that the GAN captures will allow it to represent unseen samples, thus the downstream classifiers better. We also pivoted to using the Fashion MNIST dataset instead as it has far more samples than the NeRF dataset and has also proven to have better results when training a GAN model. This resulted in the final setup that we went with for our project. We first artificially unbalanced the Fashion MNIST dataset. We will refer to this dataset as $\mathcal{D}_{UB}$. $\mathcal{D}_{UB}$ was created by iterating over the classes. At each iteration i, the number of samples was set to be $C \times prevNumSamples$ where $prevNumSamples$ is the number of samples of class $i-1$ and $0 < C < 1$ is some unbalancing ratio. Note that the lower C is, the more long-tailed the dataset will be.

We then trained a conditional GAN on an upsampled $\mathcal{D}_{UB}$ and used the trained model to generate new samples for each of the classes in the Fashion MNIST dataset. We upsampled the data the GAN was trained on so that the GAN could learn representations of classes with few samples.

The architecture for the basic GAN model that we created is shown in Figure 1 below. The generator is composed of linear layers with Leaky ReLU activations. Other Generator architectures will be discussed in the experiments that were used to evaluate performance on different architectures. The discriminator architecture was held constant in all GAN models that were trained.

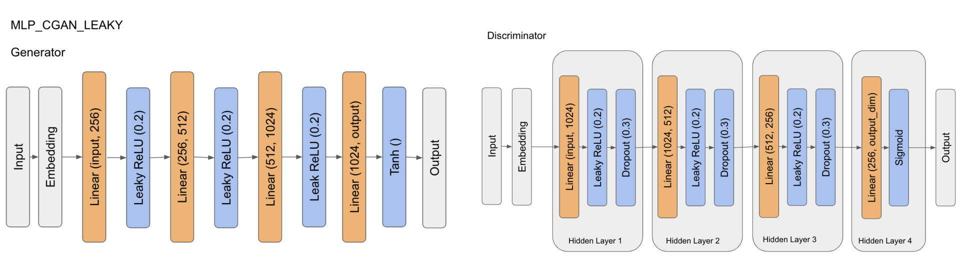


Figure 1: Generator/Discriminator Architectures for `MLP_CGAN_LEAKY`

Using the synthetic samples that were created by the GAN, we augmented the original unbalanced dataset to create a new dataset $\mathcal{D}_B$ which is balanced across the different classes. Our hypothesis was that if $\mathcal{D}_{UB}$ and $\mathcal{D}_B$ were used as training inputs to a convolutional network to perform an image classification task, the latter would produce better results. We trained the CNN using the rebalanced data coming from baselines or a GAN model and then tested the accuracy of these models on a test dataset consisting of the original balanced Fashion MNIST dataset. We chose to evaluate the models and baselines using the original Fashion MNIST test set because we wanted a consistent way to compare models that were trained on different datasets, and we wanted to view how well the models would perform on the long-tail classes.

In the experiments described below, we tested this hypothesis. We also trained several variants of the CGAN architecture described above to see which would lead to the best performance against the baselines of the unbalanced dataset, the dataset balanced with RU, and the dataset balanced with RO.

3.1 Experiment 1

In this experiment, we compared the results of the CNN on 5 datasets.

- Original balanced Fashion MNIST dataset
- Unbalanced Fashion MNIST dataset ($C = 7/9$)
- Dataset rebalanced using RU
- Dataset rebalanced using RO
- Dataset rebalanced with synthetic samples from `MLP_CGAN_LEAKY` GAN model

The training process for the CGAN architectures consisted of training for 100 epochs with a batch size of 256 and using the Adam optimizer with a learning rate of 0.0002 and $\beta_1 = 0.9$ and $\beta_2 = 0.999$ by alternating the update of the discriminator and the generator the original conditional GAN paper [2].

The training process for the Convolutional Neural Networks consisted of training for 30 epochs (which was sufficient for convergence), a batch size of 100, and using an Adam optimizer with a learning rate of 0.0005 and $\beta_1 = 0.9$ and $\beta_2 = 0.999$. This procedure is maintained for future experiments and the only things that are modified are the CGAN architectures and datasets.

3.2 Experiment 2

In this experiment, we trained 3 more GAN models to determine whether the architecture of the GAN significantly impacted the quality of synthetic samples that were generated and therefore impacted the accuracy of the convolutional model that was trained. The datasets that were compared in this experiement were the 5 from experiment 1 along with 3 new ones listed below.

- Rebalancing with synthetic samples from `BIG_MLP_CGAN_LEAKY` (Figure 2)
- Rebalancing with synthetic samples from `SMALL_MLP_CGAN_LEAKY` (Figure 3a)
- Rebalancing with synthetic samples from `MLP_CGAN_RELU` (Figure 3b)

The architectures of these three new GANS are shown below. Note that only the architecture of the generator was changed, the discriminator used had the same architecture as shown in Figure 1 above.

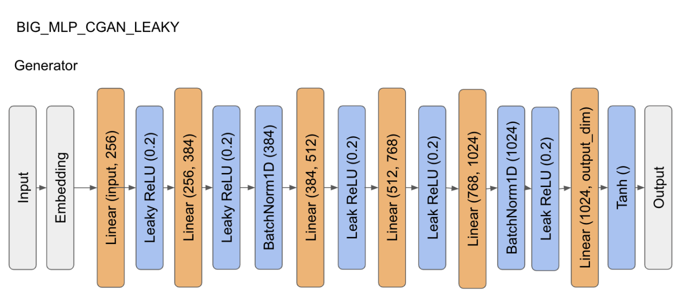
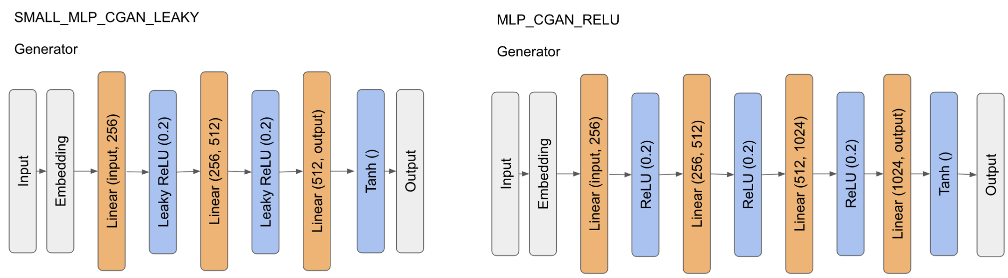


Figure 2: `BIG_MLP_CGAN_LEAKY` GAN model



(a) `SMALL_MLP_CGAN_LEAKY` GAN model (b) `MLP_CGAN_RELU` GAN model

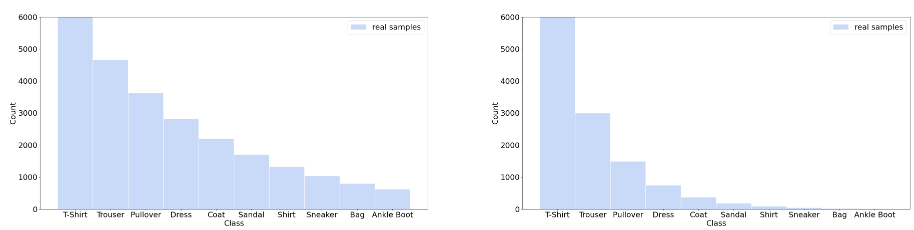
Figure 3: `SMALL_MLP_CGAN_LEAKY` and `MLP_CGAN_RELU` GAN models

3.3 Experiment 3

In our final experiment, we looked at how more significant class imbalance would impact the accuracy of image classification in the end. In order to do this, we chose two different values of C: $7/9$ and $1/2$. Recall that the value of C was used to decrease the number samples over the different classes (if class 1 had 100 samples and $C = 1/2$, class 2 would have 50 samples, class 3 would have 25 and so on).

Using these two values of C, we now have two unbalanced datasets, call these $\mathcal{D}_{UB\_low}$ for the dataset created by setting $C = 7/9$ and $\mathcal{D}_{UB\_high}$ for the dataset created by setting $C = 1/2$. We then trained all of the GAN models that were trained in experiment 2 using both of these unbalanced datasets. The trained models were used to create rebalanced datasets for each.

The data distributions for the unbalanced datasets are visualized below.



(a) Distribution for $\mathcal{D}_{UB\_low}$ (b) Distribution for $\mathcal{D}_{UB\_high}$

4 Results

Table 1 below lists the losses and accuracies of all the methods listed in the experiments sections.

4.1 Experiment 1

In this experiment, we compared the results of using the baseline datasets versus the dataset that was rebalanced using `MLP_CGAN_LEAKY GAN` model for training of the CNN. Validation was done using the full Fashion MNIST dataset.

Figure 5 below are some synthetic samples that were produced by the `MLP_CGAN_LEAKY GAN` model. Samples such as these were used to rebalance the unbalanced Fashion MNIST dataset.

Dataset	Rebalancing Modification	Final Loss	Final Accuracy	Final F1
original	—	1.540	0.921	0.921
low_unbalanced	—	1.571	0.890	0.884
low_unbalanced	downsampled	1.584	0.876	0.873
low_unbalanced	upsampled	**1.561**	**0.900**	**0.898**
low_unbalanced	mlp_cgan_relu	1.585	0.876	0.872
low_unbalanced	small_mlp_cgan_leaky	1.574	0.887	0.882
low_unbalanced	mlp_cgan_leaky	1.670	0.791	0.751
low_unbalanced	big_mlp_cgan_leaky	1.586	0.875	0.868
high_unbalanced	—	1.983	0.476	0.342
high_unbalanced	downsampled	1.763	0.698	0.670
high_unbalanced	upsampled	**1.648**	**0.812**	**0.799**
high_unbalanced	mlp_cgan_relu	1.727	0.733	0.715
high_unbalanced	small_mlp_cgan_leaky	1.775	0.686	0.658
high_unbalanced	mlp_cgan_leaky	1.766	0.696	0.668
high_unbalanced	big_mlp_cgan_leaky	1.806	0.654	0.623

Table 1: Final Results of Combined Experiments

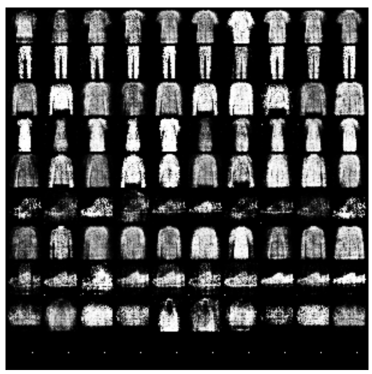


Figure 5: Samples generated from `MLP_CGAN_LEAKY` model

Using these samples, we created a rebalanced dataset. These datasets were then used to train a CNN model on an image classification task. The test dataset was the full Fashion MNIST dataset. Below are the test loss, test accuracy, and test F1 scores of the baseline models as well as our new approach.

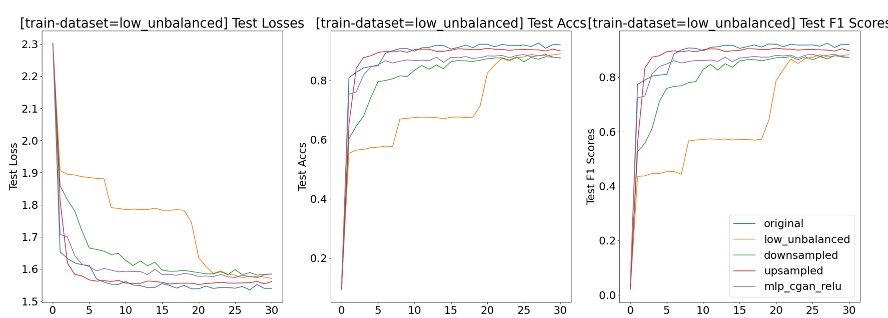


Figure 6: Testing Loss, Accuracy, and F1 Scores on CNNs generated by different methods

4.2 Experiment 2

In this experiment, we compared the results of using the datasets from experiment 1 as well as 3 new GAN architectures that we trained. Validation was done using the full Fashion MNIST dataset.

Below are some synthetic samples that were produced by the three new GAN models. Samples such as these were used to rebalance the unbalanced Fashion MNIST dataset before training the CNN.

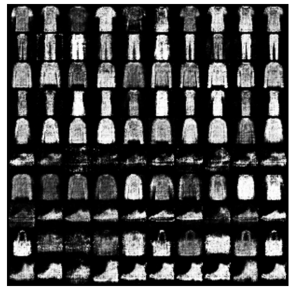
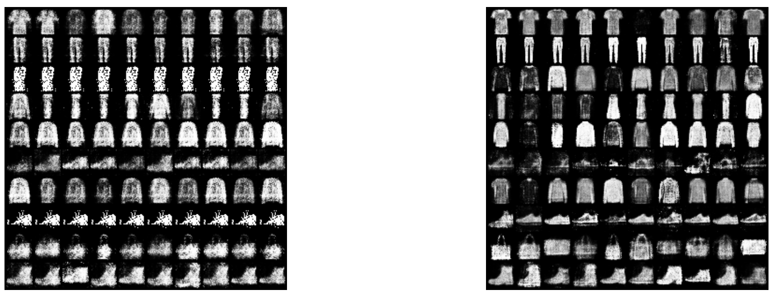


Figure 7: Samples from `SMALL_MLP_CGAN_LEAKY`



(a) Samples from `BIG_MLP_CGAN_LEAKY`

(b) Samples from `MLP_CGAN_RELU`

Figure 8: Samples from `BIG_MLP_CGAN_LEAKY` and `MLP_CGAN_RELU`

Using these samples, we created three more rebalanced datasets. These datasets were then used to train a CNN model on an image classification task. The test dataset was the full Fashion MNIST dataset. Below are the test loss, test accuracy, and test F1 scores of the baseline models as well as the GANs that we trained.

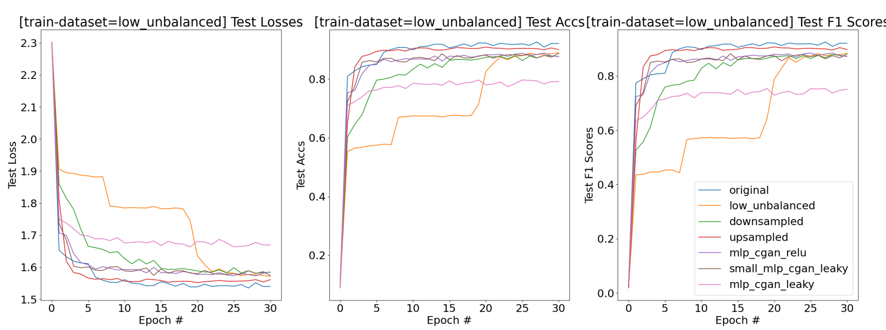


Figure 9: Testing Loss, Accuracy, and F1 Scores on CNNs generated by different methods

4.3 Experiment 3

In this experiment, we repeated the training that was done in Experiment 2 for another dataset that had a higher amount of imbalance ($C = 1/2$) as compared to $C = 7/9$. The goal of this experiment was to determine the impact of the severity of the original imbalance on the accuracy of the CNN in the end.

All 4 GAN architectures that have been described above in this paper were trained again using the $\mathcal{D}_{UB\_high}$ dataset (described in the methods section). Below are some samples that were generated by one of these models.

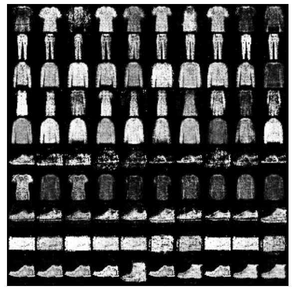


Figure 10: Samples from `MLP_CGAN_LEAKY` on $\mathcal{D}_{UB\_high}$

Rebalanced datasets were then created in the same method described already using synthetic samples from these 4 trained models to create 4 training datasets that were used to train the CNN model. The results this along with the baseline methods is shown below.

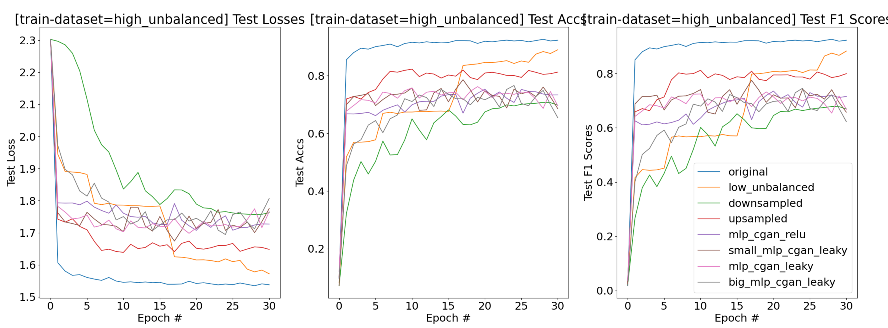


Figure 11: Testing Loss, Accuracy, and F1 Scores on CNNs generated by different methods

5 Discussion and Analysis

5.1 Experiment 1

After training the various models and running the experiments described in Experiment 1, we found that the upsampling method produced the best performance, closest to the optimal performance set by training using the original dataset; however, we also found that the synthetic GAN-datasets still proved to produce rates and improvement on performance metrics in comparison to random undersampling, despite performing sub-optimally in comparison to random oversampling.

One reason that we think that random oversampling may have performed better than the conditional GAN is because of the nature of the Fashion MNIST dataset. Because samples from a given class do not have much variation between them (not much detail in the images of the clothing), simply copying an image that is already in the dataset may prove better than generating a synthetic image from our GAN that may not be fit the true distribution of the class samples and instead introduce significant noise into the dataset.

However, compared to the other baseline method of random undersampling, using synthetic data from the GAN performed quite well. This is because random undersampling loses a lot of information from the original dataset by throwing away samples. By rebalancing the dataset by adding to synthetic samples to the classes with fewer samples, we are not losing information from the original dataset.

5.2 Experiment 2

Among the three different GAN architectures we tested, the `SMALL_MLP_CGAN_LEAKY` and `MLP_CGAN_RELU` GAN generators were comparable and performed better than the `MLP_CGAN_LEAKY` model. While we initially believed that LeakyReLU would have improved performance, we believe that we received these results due to the fact that the FashionMNIST dataset is not a very complicated data distribution, in comparison to other image datasets like CIFAR10. As a result, we believe that the smaller generative model performed better because it didn't overfit as strongly to the dataset as did the other models, such as the `MLP_CGAN_LEAKY`.

5.3 Experiment 3

As we vary the degree of imbalance, we see that a lower imbalance constant $C = 1/2$ causes the losses of all the models to decrease. In addition, the convergence rates are noticeably less stable compared to the models from the less imbalanced datasets. Because the dataset is more long-tailed, the number of synthetic examples for the less-represented image classes increases dramatically. These synthetically generated images are naturally more correlated with the already existing images in the class. Therefore, we introduce a larger intra-class variability and small inter-class variability into our dataset. Images in each dataset become more similar, and images between datasets become less similar. This leads to features becoming more distinct. We believe this variability causes the losses of our models to fluctuate much more and converge to a lower value.

5.4 Limitations and Conclusions

Our experiment on the Fashion MNIST dataset is limited by the simplicity and uniformity of the dataset. Instead of working with a naturally unbalanced dataset, we had to manually carve out long-tailed distributions in order to artificially generate new samples. The simplicity of our model led to better relative performance of simpler re-balancing techniques like oversampling. Our GAN models would likely benefit more from a naturally unbalanced and richer dataset such as CIFAR10 or our original NeRF dataset. The correlation and interaction between features would be much more complex, and it would be interesting to see how methods like upsampling and GAN generation change the variability of these features. Perhaps we would need more complex re-balancing methods and more fine-tuned GAN architectures.

Furthermore, we found that we were working with a dataset with only 10 classes. In practice, many larger-scale image datasets have a higher number of classes ranging from hundreds to thousands. In these datasets, the imbalance problem becomes more significant and likely can not be solved by simply random oversampling, as was shown in this paper with the Fashion MNIST dataset.

Our results overall show that GANs are not necessarily helpful in simple data distributions, despite the high-dimensionality of image data. Instead, simple and more efficient methods such as over-sampling can help deal with the data-imbalance problem and still gain strong performance in comparison to training on unbalanced datasets. Future work should consider attempting these re-balancing methods on datasets that have follow more complex distributions and have a greater number of classes as well. While our work suggests that GANs do not add useful performance gains in improving rebalancing datasets, it is imperative to try these methods in more complex and extreme datasets that are more likely to occur in the wild.

5.5 Code

Training code, models, datasets, and methods can be found in the following Github Repository (https://github.com/dinodeep/10707-Project).

References

[1] Maureen Lyndel C. Lauron and Jaderick P. Pabico. Improved sampling techniques for learning an imbalanced data set. *CoRR*, abs/1601.04756, 2016.

[2] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[3] Vignesh Sampath, Iñaki Maurtua, J. Aguilar, and Aitor Gutierrez. A survey on generative adversarial networks for imbalance problems in computer vision tasks, 07 2020.