



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JEFFERSON LUIZ PESSOA LIMA

**ADVERSARIAL OVERSAMPLING: Um método para balanceamento baseado
em Redes Neurais Adversárias**

Recife
2021

JEFFERSON LUIZ PESSOA LIMA

ADVERSARIAL OVERSAMPLING: Um método para balanceamento baseado em Redes Neurais Adversárias

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Orientador: Prof. Paulo Salgado Gomes de Mattos Neto

Coorientador: Prof. Fernando José Ribeiro Sales

Recife
2021

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

L732a Lima, Jefferson Luiz Pessoa
Adversarial oversampling: um método para balanceamento baseado em redes neurais adversárias / Jefferson Luiz Pessoa Lima. – 2020.
75 f.: il., fig., tab.

Orientador: Paulo Salgado Gomes de Mattos Neto.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2020.
Inclui referências.

1. Inteligência computacional. 2. Redes neurais. I. Mattos Neto, Paulo Salgado Gomes de (orientador). II. Título.

006.31

CDD (23. ed.)

UFPE - CCEN 2020 - 209

JEFFERSON LUIZ PESSOA LIMA

**“ADVERSARIAL OVERSAMPLING: UM MÉTODO PARA
BALANCEAMENTO BASEADO EM REDES NEURAS ADVERSÁRIAS”**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 29/09/2020.

BANCA EXAMINADORA

Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPE

Prof. Dr. Leandro Augusto da Silva
Faculdade de Computação e Informática / Mackenzie

Prof. Dr. Paulo Salgado Gomes de Mattos Neto
Centro de Informática / UFPE
(Orientador)

Dedico esta dissertação a toda minha família, aos amigos e professores que me deram o apoio necessário para chegar até aqui.

AGRADECIMENTOS

Primeiramente, agradeço a Deus e minha família pela força e motivação, fundamentais para superar os desafios enfrentadas no decorrer da pós graduação. Tornando possível superar todos os obstáculos que surgiram, sem nem ao menos pensar em desistir.

Agradeço ao apoio financeiro concedido pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), que tornaram possível arcar com as despesas necessárias para conclusão do curso. Além de agradecer a Universidade Federal de Pernambuco (UFPE) e o Centro de Informática (Cin), que me forneceu uma excelente oportunidade com uma ótima estrutura e um adequado ambiente de aprendizado.

Agradeço a todos os meus amigos e colegas que passaram por mim durante o curso e que acrescentaram de forma direta e indireta ao longo das disciplinas e projetos.

Agradeço aos extremamente capacitados professores das disciplinas cursadas, que contribuíram imensamente para meu conhecimento com suas aulas e ensinamentos.

Agradeço, por fim, aos orientadores/professores, Paulo Salgado e Fernando Sales, que foram de suma importância para meu desenvolvimento acadêmico, contribuindo com dicas valiosas. Com extrema paciência, me guiaram da melhor forma para uma proveitosa pós-graduação.

RESUMO

Base de dados com diferentes proporções de exemplos por classe é algo frequente e pode afetar negativamente alguns algoritmos de aprendizagem supervisionada. Uma série de classificadores podem ficar enviesados quando são treinados nessas condições, especializando-se somente nas classes com maior quantidade de amostras, deixando de lado as classes minoritárias, as quais podem ser mais importantes para o objetivo em questão. Esse problema exige um esforço considerável de investigação e estudo para elaboração de métodos. Como nem sempre é possível, ou viável, a coleta de novas amostras da classe minoritária, métodos têm sido estudados e desenvolvidos para realizar a geração de amostras a partir das existentes, fazendo o chamado *Oversampling*. Utilizando as *Generative Adversarial Networks* - GANs, as quais compõem o estado da arte para geração de amostras sintéticas realistas, o presente trabalho propôs o método *Adversarial Oversampling* - AO. O método proposto utiliza as *Auxiliary Classifier GANs* - AC-GAN como rede adversária, para aprender a distribuição das classes e gerar amostras sintéticas para balancear a base de dados. Durante o treinamento do AO, é dada uma maior atenção às amostras consideradas difíceis, utilizando a própria rede adversária para determinar a dificuldade dessas amostras. Com isso, o AO tenta gerar amostras mais significativas, que colaboram para melhorar a região de fronteira do classificador em questão. As amostras sintéticas geradas são filtradas utilizando a rede adversário do AO. A primeira análise realizada teve como objetivo observar a morfologia das amostras geradas pelo *Oversampling*, e mostrou que o AO conseguiu gerar amostras com um aspecto visual mais fiel às amostras originais, comparado aos métodos tradicionais de *Oversampling*, que possuem a geração baseada em interpolação. Uma segunda análise foi feita para observar a precisão do classificador na classe minoritária à medida que são inseridas as amostras geradas pelo AO. Por fim, foi feito um experimento comparativo entre os métodos de *Oversampling*, observando o quão as amostras geradas contribuem para o aumento na precisão do classificador na classe minoritária. Esse experimento foi repetido em cinco bases de dados de imagens. Os resultados médios do AO mostraram indícios de superioridade em quatro das cinco das bases, em até 4%. Com base nos indícios obtidos através dos experimentos realizados, pode-se observar que o método proposto pode ser capaz de gerar amostras sintéticas, balanceando a distribuição das classes e contribuindo no processo de aprendizagem do classificador.

Palavras-chave: Oversampling. Desbalanceamento. Redes Neurais Adversarias. Gans.

ABSTRACT

Datasets with different proportions of examples per class are common and negatively affect some supervised learning algorithms. Some classifiers can be biased when trained in these conditions, specializing only in classes with a greater number of samples, leaving aside minority classes, which may be more important for the objective in question. This problem requires considerable research and study effort to develop methods. How is not always possible, or feasible, to collect new samples from the minority class, the methods have been studied for generating samples from existing ones. Using the *Generative Adversarial Networks - GANs*, which are the state of the art for generating realistic synthetic samples, the present work proposed the *Adversarial Oversampling - AO* method. The proposed method use the *Auxiliary Classifier GANs - AC-GAN* as an opposing network, to learn the distribution of classes and generate synthetic samples to balance the dataset, focusing on the minority class. During the training of *AO*, a preference is given to hardest samples, using the Discriminator **D** network to determine the difficulty of these samples. Further *AO* tries to generate more significant samples, which collaborate to improve the decision region of the classifier. The generated synthetic samples are filtered using the *AO* adversary network. The first analysis performed aimed to observe the morphology of the images samples generated by *Oversampling* and showed that *AO* generate samples with a visual aspect more faithful to the original samples, compared to traditional methods of *Oversampling*, which have an interpolation-based generation. A second analysis was made to observe the classifier's accuracy in the minority class as the samples generated by *AO* are inserted. Finally, a comparative experiment was performed between the *Oversampling* methods, observing how the generated samples contribute to the classifier accuracy in the minority class. This experiment was repeated in five image datasets. The average results of *AO* showed evidence of superiority in four of the five datasets, up to 4%. Based on the evidence obtained through the experiments, the proposed method may be able to generate synthetic samples, balancing the the classes proportion, and contributing to the learning process of the classifier.

Keywords: Oversampling. Class Imbalance. Generative Adversarial Networks. Gans.

LISTA DE FIGURAS

Figura 1 – Ilustração do funcionamento do Gerador G , o qual recebe um ruído \mathbf{z} e gera uma amostra \mathbf{X}_{fake} , e do Discriminador D , recebe \mathbf{X}_{fake} e \mathbf{X} e as classifica em real ou falso (0 ou 1)	24
Figura 2 – Arquitetura geral das GANs	24
Figura 3 – Arquitetura geral da CGANs	26
Figura 4 – Ilustração do funcionamento do Gerador G , o qual recebe um ruído \mathbf{z} , uma informação condicional c e gera uma amostra \mathbf{X}_{fake} condicionada à c , e do Discriminador D , que recebe \mathbf{X}_{fake} , \mathbf{X} e c e classifica-os em real ou falso (0 ou 1) de forma condicionada à c .	26
Figura 5 – Arquitetura da AC-GAN, onde D é o Discriminador, G o Gerador e c e \mathbf{z} são a classe e o ruído, respectivamente.	27
Figura 6 – Amostras sintéticas geradas pelo SMOTE, para a classe minoritária representada por \bullet . Em verde as amostras sintéticas e em vermelho as amostras reais, utilizadas na geração.	32
Figura 7 – Exemplo da aplicação do SVM SMOTE. No lado esquerdo ocorre a extrapolação, onde a amostra sintética (verde) extrapola a região entre os exemplos minoritários (azul). Já no lado direito, ocorre a interpolação, onde a amostra sintética (verde) é gerada entre os exemplos minoritários (azul).	34
Figura 8 – Fluxo do uso <i>Adversarial Oversampling</i> para balancear o a base de dados \mathbf{X}	37
Figura 9 – Arquitetura geral do <i>Adversarial Oversampling</i>	38
Figura 10 – Arquitetura das Redes Adversarias utilizadas como base do <i>Adversarial Oversampling</i> . No lado esquerdo é exibido à estrutura do Gerador G e do Discriminador D , o qual contém as camadas de ruído gaussiano e <i>Minibatch-Discrimination</i> . No lado direito, está o fluxo de treinamento, seguindo o padrão das AC-GAN	39
Figura 11 – Plot 2D de amostras, visualmente similares, dos dígitos 4 e 9. . . .	44
Figura 12 – Bloco H que representa a etapa de Entropia para determinar a dificuldade das amostras.	44

Figura 13 – Bloco que representa a etapa de filtragem das amostras sintéticas geradas pelas Redes Adversárias. Recebendo um <i>batch</i> de amostras sintéticas \mathbf{X}_s e o Discriminador \mathbf{D} , já treinado. Como saída, \mathbf{X}'_s são as imagens que passaram pela filtragem. Consistindo na etapa final do <i>Adversarial Oversampling</i>	44
Figura 14 – Exemplo de imagens da base de dados <i>Breast</i>	46
Figura 15 – Exemplo de imagens da base de dados <i>XRy</i>	46
Figura 16 – Exemplo de imagens da base de dados <i>Skin Cancer</i>	47
Figura 17 – Exemplo do sinal de ECG da base de dados <i>Heartbeat</i>	47
Figura 18 – Exemplo de imagens da base de dados <i>Heartbeat</i> , das quatro classes existentes, após pré-processamento.	48
Figura 19 – Exemplo de imagens da base de dados <i>Glaucome</i>	48
Figura 20 – Exemplo da área sobre a curva. No eixo x há o <i>precision</i> e no eixo y o <i>recall</i>	50
Figura 21 – Fluxo experimental da comparação entre o método proposto e os demais métodos de <i>Oversampling</i>	53
Figura 22 – Geração de amostras sintéticas dos dígitos 0 e 4 utilizando os métodos tradicionais (esquerda) e utilizando o AO (direita).	55
Figura 23 – Geração de amostras sintéticas de imagens da base <i>Skin Cancer</i> utilizando os métodos tradicionais (esquerda) e utilizando o AO (direita).	56
Figura 24 – Geração de amostras sintéticas de imagens da base <i>Heartbeat</i> utilizando os métodos tradicionais (esquerda) e utilizando o AO (direita).	56
Figura 25 – Geração de amostras sintéticas de imagens da base <i>XRy</i> utilizando os métodos tradicionais (esquerda) e utilizando o AO (direita).	57
Figura 26 – Variação da precisão da classe minoritária na base <i>XRy</i> , à medida que são inseridas amostras sintéticas geradas pelo AO.	58
Figura 27 – Variação da precisão da classe minoritária na base <i>Skin Cancer</i> , à medida que são inseridas amostras sintéticas geradas pelo AO.	58
Figura 28 – Variação da precisão da classe minoritária 2 na base <i>Heartbeat</i> , à medida que são inseridas amostras sintéticas geradas pelo AO.	59
Figura 29 – Variação da precisão da classe minoritária 3 na base <i>Heartbeat</i> , à medida que são inseridas amostras sintéticas geradas pelo AO.	59
Figura 30 – Variação da precisão da classe minoritária na base <i>Glaucome</i> , à medida que são inseridas amostras sintéticas geradas pelo AO.	60

Figura 31 – Variação da precisão da classe minoritária na base <i>Breast</i> , à medida que são inseridas amostras sintéticas geradas pelo AO. . . .	60
Figura 32 – Variação da precisão da classe minoritária na base <i>XRay</i> , à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.	61
Figura 33 – Variação da precisão da classe minoritária na base <i>Skin Cancer</i> , à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem	62
Figura 34 – Variação da precisão da classe minoritária 2 na base <i>Heartbeat</i> , à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.	62
Figura 35 – Variação da precisão da classe minoritária 3 na base <i>Heartbeat</i> , à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.	63
Figura 36 – Variação da precisão da classe minoritária na base <i>Glaucome</i> , à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.	63
Figura 37 – Variação da precisão da classe minoritária na base <i>Breast</i> , à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.	64

LISTA DE TABELAS

Tabela 1 – Sumarização dos métodos de <i>Oversampling</i> apresentados	35
Tabela 2 – Sumarização das bases de dados utilizadas	49
Tabela 3 – Infra-estrutura utilizada para realização dos experimentos.	53
Tabela 4 – Resultados da métrica <i>Gmean</i> nas bases de dados utilizando o classificador VGG16.	65
Tabela 5 – Resultados da métrica AUC-ROC nas bases de dados utilizando o classificador VGG16.	65
Tabela 6 – Resultados da métrica <i>Gmean</i> nas bases de dados utilizando o classificador ResNet.	66
Tabela 7 – Resultados da métrica AUC-ROC nas bases de dados utilizando o classificador ResNet.	66
Tabela 8 – Resultados da métrica <i>Gmean</i> nas bases de dados utilizando o classificador <i>DenseNet</i>	66
Tabela 9 – Resultados da métrica AUC-ROC nas bases de dados utilizando o classificador <i>DenseNet</i>	67

LISTA DE ABREVIATURAS E SIGLAS

AC-GAN	<i>Auxiliary Classifier GANs</i>
ADASYN	<i>Adaptive Synthetic</i>
AO	<i>Adversarial Oversampling</i>
AUC-ROC	<i>Area Under the Curve - ROC</i>
BN	<i>Batch Normalization</i>
BSMOTE	<i>Borderline SMOTE</i>
CGANs	<i>Conditional Generative Adversarial Networks</i>
CNN	<i>Convolutional Neural Networks</i>
GANs	<i>Generative Adversarial Networks</i>
KSMOTE	<i>K-Means SMOTE</i>
Leaky-ReLU	<i>Leaky Rectified Linear Unit</i>
ReLU	<i>Rectified Linear Unit</i>
ROS	<i>Random Oversampling</i>
SMOTE	<i>Synthetic Minority Over-sampling Technique</i>
SVMSMOTE	<i>SVM-SMOTE</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS	18
1.2	CONTRIBUIÇÕES	18
1.3	ESTRUTURA DA DISSERTAÇÃO	18
2	REFERENCIAL TEÓRICO	21
2.1	DESBALANCEAMENTO EM BASES DE DADOS	21
2.1.1	<i>Undersampling</i>	22
2.1.2	<i>Oversampling</i>	23
2.2	GENERATIVE ADVERSARIAL NETWORKS - GANS	23
2.2.1	<i>Conditional Generative Adversarial Networks - CGAN</i>	25
2.3	AUXILIARY CLASSIFIER GANS - AC-GAN	26
2.4	CAMADA DE RUÍDO GAUSSIANO	27
2.5	MINIBATCH DISCRIMINATION	28
2.6	FILTRAGEM DE DADOS SINTÉTICOS GERADOS POR GANS	29
2.6.1	<i>Class conditional probability</i>	29
2.6.2	<i>Realism conditional probability</i>	29
3	TRABALHOS RELACIONADOS	31
3.1	RANDOM OVERSAMPLING - ROS	31
3.2	SYNTHETIC MINORITY OVERSAMPLING - SMOTE	31
3.3	ADAPTIVE SYNTHETIC - ADASYN	32
3.4	BORDERLINE-SMOTE	33
3.5	SVMSMOTE	34
3.6	KMEANS-SMOTE	35
4	MODELO PROPOSTO	36
4.1	ADVERSARIAL OVERSAMPLING - AO	36
4.1.1	Utilizando o Adversarial Oversampling	36
4.1.2	Arquitetura Geral do Modelo	37
4.1.3	Arquitetura das Redes Adversárias	37
4.1.3.1	Rede Geradora (G)	38
4.1.3.2	Rede Discriminadora (D)	39
4.1.4	Etapas do Modelo	41
4.1.4.1	Treinamento Inicial	41

4.1.4.2	Segundo Treinamento - Especializando o Gerador	41
4.1.4.3	Filtragem das Amostras Sintéticas	43
5	EXPERIMENTOS	45
5.1	BASES DE DADOS	45
5.1.1	<i>Breast</i>	45
5.1.2	<i>XRay</i>	45
5.1.3	<i>Skin Cancer</i>	46
5.1.4	<i>Heartbeat - MITBIH</i>	47
5.1.5	<i>Glaucome - ORIGA</i>	48
5.2	MEDIDAS DE AVALIAÇÃO	49
5.2.1	Média Geométrica	49
5.2.2	AUC-ROC	50
5.3	DIVISÃO DOS CONJUNTOS DE DADOS	50
5.4	CLASSIFICADORES	51
5.5	AVALIAÇÕES	51
5.5.1	Contribuição do AO para o Acerto da Classe Minoritária . . .	51
5.5.2	Impacto da Filtragem no AO	52
5.5.3	Aspecto Visual das Amostras	52
5.5.4	Comparativo entre os Métodos Tradicionais de <i>Oversampling</i>	52
5.6	AMBIENTE	53
5.7	CONFIGURAÇÃO PARAMÉTRICA PARA COMPARAÇÃO DOS MÉTODOS .	54
6	RESULTADOS E DISCUSSÕES	55
6.1	QUALIDADE VISUAL	55
6.2	AMOSTRAS GERADAS PELO AO E PRECISÃO	57
6.3	IMPACTO DA FILTRAGEM NO AO	61
6.4	MÉTODOS TRADICIONAIS <i>vs</i> AO	64
6.4.1	Resultados utilizando o classificador VGG16	65
6.4.2	Resultados utilizando o classificador ResNet	65
6.4.3	Resultados utilizando o classificador DenseNet	66
6.4.4	Análise dos Resultados	67
7	CONCLUSÕES	68
7.1	TRABALHOS FUTUROS	69
	REFERÊNCIAS	70

1 INTRODUÇÃO

Com o passar do tempo, pode-se notar cada vez mais o crescimento do uso de Aprendizagem de Máquina na automação de diversas tarefas, dentre as quais estão relacionadas o processamento de linguagem natural (DEVLIN et al., 2018), predição de séries temporais (SEZER; GUDELEK; OZBAYOGLU, 2019) e classificação de imagens (Sultana; Sufian; Dutta, 2018). Essa última, faz parte do campo de Visão Computacional, e pode ser definida como a tarefa de categorização de uma imagem em uma ou mais classes (Nath et al., 2014).

Para da tarefa de classificação de imagens, por muito tempo, foi utilizada uma abordagem de dois níveis: extração de características e classificação (TIAN, 2013). Essa abordagem torna o processo muito dependente da qualidade da etapa de extração de características, a qual exige um certo conhecimento na área de domínio do problema. As Redes Neurais Convolucionais (do inglês: *Convolutional Neural Networks* - *CNNs*) foram propostas com o objetivo de automatizar o processo da extração de características, através de transformações não-lineares (LECUN et al., 1999). Isso caracterizou o chamado Aprendizado Profundo (do inglês: *Deep Learning*, que consiste no uso de Redes Neurais profundas, com múltiplas camadas, para extração e classificação de características (RAWAT; WANG, 2017), (LU; WENG, 2007), (Sornam; Muthusubash; Vanitha, 2017).

Um problema comum, envolvendo a forma supervisionada da Aprendizagem de Máquina, é o desbalanceamento de classes. Trazendo para o contexto do presente trabalho, esse problema ocorre quando existe um desequilíbrio na quantidade de imagens por classes. As classes com maior quantidade de exemplos são chamadas de majoritárias, e as demais, com menor representação, são chamadas de minoritárias. O desequilíbrio na representação das classes pode afetar negativamente o treinamento de aprendizagem supervisionada, como as Redes Neurais profundas. Esses modelos podem gerar um favorecimento à classe com maior representação na base de dados, e um menor favorecimento as classe com menos amostras. Um exemplo de modelo amplamente utilizado hoje em dia, e que sofre quando são treinados sobre dados desbalanceados, são as Redes Neurais Profundas, incluindo as *Convolutional Neural Networks* (CNN). Isso ocorre porque, durante o treinamento, os neurônios são ajustados mais vezes de acordo com as amostras da classe mais frequente da base de dados. Com isso, a superfície de decisão criada é inadequada, pois ignora classe minoritária. Isso pode fazer com que a rede tenha alta taxa de acerto na classe majoritária, porém uma taxa inaceitável na classe minori-

tária (JOHNSON; KHOSHGOFTAAR, 2019).

Os problemas causados por desbalanceamento de classes são objetos de pesquisas em busca de soluções para acabar, ou pelo menos reduzir, os efeitos negativos causados durante o treinamento. Os métodos desenvolvidos para lidar com esse problema são agrupados em três: No nível de dados, que trata a base de dados realizando alguma re-amostragem a fim de equilibrar a proporção das classes, por exemplo, gerando exemplos artificiais para a classe minoritária (CHAWLA et al., 2002); No nível de algoritmo, que ajusta o modelo durante o treinamento, por exemplo, adicionando uma equidade entre as classes no processo de ajuste dos pesos numa rede neural; E a combinação dos métodos a nível de dados e a nível de algoritmo. A nível de dados, existem os métodos que aumentam a quantidade de amostras minoritárias, chamados de *Oversampling*, e os métodos que diminuem a quantidade de amostras das classes majoritárias, chamados de *Undersampling* (He; Garcia, 2009), (YANMINSUN; WONG; KAMEL, 2011), (KRAWCZYK, 2016). O presente trabalho se concentra na abordagem à nível de dados, mais especificamente, nos métodos de *Oversampling*. Essa escolha se pois, com o uso de *Oversampling*, se tem um aproveitamento completo de todas as amostras, utilizando assim toda a informação disponível na base de dados, sem perder uma informações úteis presentes nas amostras da classe majoritária.

Para a tarefa de *Oversampling*, o método mais simples é o *Random Oversampling - ROS*, no qual ocorre o processo de repetição aleatória das amostras das classes minoritárias. Porém, métodos que realmente criam amostras foram desenvolvidos, dentre eles: SMOTE (CHAWLA et al., 2002), ADASYN (Haibo He et al., 2008), Borderline-SMOTE (HAN; WANG; MAO, 2005), KMeans-SMOTE (LAST; DOUZAS; BAÇÃO, 2017) e SVM-SMOTE (NGUYEN; COOPER; KAMEL, 2011). Esses métodos, explicados no Capítulo 2, são versões modernas do SMOTE. Utilizam interpolação e/ou extrapolação das amostras existentes, para geração das amostras sintéticas. Essas variações permanecem amplamente utilizadas e ainda são estudadas (Abdoh; Abo Rizka; Maghraby, 2018), (Saad; Omar; Maghraby, 2019), (Mathew et al., 2018), (Ghosh et al., 2019), (Satriaji; Kusumaningrum, 2018), (Demidova; Klyueva, 2017), (Zhang; Guo; Lu, 2017).

Um cuidado a ser tomado durante a prática de *Oversampling* é garantir que a amostra gerada para a classe minoritária seja representativa ao ponto de trazer informações sobre a mesma. Caso contrário, essa amostra gerada pode não contribuir para o aprendizado ou, até mesmo, o atrapalhar.

Quando o assunto é geração de imagens, temos novamente a presença de *Deep Learning* como destaque através das Redes Neurais Adversárias (*Generative Adversarial Networks* (GANs) (GOODFELLOW et al., 2014), as quais mostram potencial

na da geração de amostras em diferentes domínios (GECER et al., 2018), (KARRAS et al., 2017), (JIN et al., 2017), (ZHU et al., 2017), (ZHANG et al., 2016), (REED et al., 2016), (MA et al., 2017), (TAIGMAN; POLYAK; WOLF, 2016), (LIU; TUZEL, 2016). A vantagem que as GANs possuem na geração de amostras sintéticas é o fato de que essa geração ocorre após o aprendizado da distribuição original dos dados, diferente dos métodos baseados no SMOTE. Esse fato impulsionou a investigação do uso de *Generative Adversarial Networks* (GANs) para geração de amostras sintéticas que possam melhorar o desempenho de um classificador, aumentando a base de dados e contribuindo para o processo de aprendizagem supervisionada.

Diversos trabalhos exploraram o uso GANs para gerar amostras em domínios específicos, para todas as classes ou somente para classe minoritária (*Oversampling*). As GANs já foram utilizadas para gerar amostras utilizadas em: classificação de batimentos cardíacos (Lima; Macêdo; Zanchettin, 2019); classificação de tráfego (VU; BUI; NGUYEN, 2017); categorização de radiografias (Salehinejad et al., 2018), (BOZORGTABAR et al., 2019); classificação da confiabilidade de engrenagens de transmissão (Li; He; Li, 2019); segmentação e categorização de imagens de ressonância magnética (Rezaei; Yang; Meinel, 2018); categorização de condição climática (Li et al., 2018); classificação de veículos (LIU; LUO; LI, 2018); segmentação de tumores cerebrais (REZAEI; YANG; MEINEL, 2019); classificação de nódulos pulmonares (Wang et al., 2019).

Observando os trabalhos citados acima, nota-se o potencial das GANs na geração de dados sintéticos para realização de *Oversampling*. Porém, é visto que são aplicações em problemas específicos, não há uma visão geral considerando mais bases de dados e também não uma comparação com os métodos tradicionais para *Oversampling*. Além disso, não há uma forma de priorizar amostras consideradas difíceis, a partir das quais podem ser geradas amostras mais úteis, que podem contribuir mais para a formação da região de decisão (HAN; WANG; MAO, 2005). Um outro fator ausente nos trabalhos citados é a presença de uma etapa de filtragem (BHATTARAI et al., 2019), importante para diminuir a geração e a inserção de ruído na base de dados, ou na classe minoritária, quando é realizado o *Oversampling*.

Tendo em vista a problemática do desbalanceamento de classes em bases de dados envolvendo imagens e o potencial das GANs para geração de dados sintéticos, o presente trabalho traz a elaboração de um método de *Oversampling* utilizando Redes Neurais Adversárias (GANs). O método proposto foi intitulado de *Adversarial Oversampling - AO*. O AO realiza a geração de amostras através do aprendizado das classes do problema, concentrando-se na geração de amostras minoritárias consideradas difíceis, que por sua vez são mais úteis para o classificador. Ao final do processo de geração, as amostras sintéticas são filtradas para priorizar a inserção

de amostras fiéis à classe minoritária na bases de dados.

1.1 OBJETIVOS

O presente trabalho tem como objetivo geral a elaboração de um método de *Oversampling* utilizando Redes Neurais Adversárias (GANs), priorizando a geração a partir amostras consideradas difíceis e aplicando uma filtragem das amostras geradas. Para isso, os seguintes objetivos específicos foram traçados:

1. Definir o modelo de GANs como Rede Adversária para geração de dados sintéticos;
2. Elaborar uma forma de definir a dificuldade das amostras, no intuito de priorizar as amostras difíceis;
3. Aplicar uma filtragem para diminuir as amostras geradas com baixa confiabilidade;
4. Levantamento de bases de dados envolvendo imagens;
5. Comparação com métodos tradicionais de *Oversampling*.

1.2 CONTRIBUIÇÕES

A principal contribuição do presente trabalho consiste na elaboração de um método para realização de *Oversampling* em bases dedados envolvendo imagens. O intitulado *Adversarial Oversampling*, que, gera amostras sintéticas com base no aprendizado considerando todas as classes do problema. Para isso, o método se baseia nas GANs como modelo de aprendizagem, modificando sua arquitetura para lidar com o problema de desbalanceamento. O método possui um *pipeline* para treinamento com foco em geração de amostras úteis para o classificador. Com isso, o *Adversarial Oversampling* tenta gerar amostras para balancear a base de dados e ajudar o classificador final a formar uma superfície de fronteira mais adequada. A ideia que serviu de base para o trabalho, surgiu a partir de um trabalho desenvolvido e publicado, abordando o *Oversampling* através de GANs, utilizando uma base desbalanceada de imagens (Lima; Macêdo; Zanchettin, 2019).

1.3 ESTRUTURA DA DISSERTAÇÃO

O presente trabalho foi estruturado em sete capítulos que serão resumidamente apresentados a seguir:

- **Capítulo 2:** É apresentado o referencial teórico, que serve de alicerce para compreensão o presente trabalho. O capítulo traz a problemática de desbalanceamento de classes e soluções com foco em *Oversampling*. Faz menções a trabalhos que juntos compõem o método proposto. São apresentadas as GANs e duas de suas versões supervisionadas: *Conditional Generative Adversarial Networks* (CGANs) e *Auxiliary Classifier GANs* (AC-GAN). Além disso, são apresentadas duas técnicas úteis para melhorar o treinamento das GANs: *Minibatch Discrimination* e Camada de Ruído Gaussiano. Por fim, é trazido um método para filtragem das amostras geradas pelas GANs. A união dessas técnicas compõe a proposta do presente trabalho, apresentada no Capítulo 4.
- **Capítulo 3:** Traz técnicas relacionadas à problemática do presente trabalho, o desbalanceamento de classes. As técnicas apresentadas são abordagens baseadas na prática de *Oversampling* e geram amostras para a classe minoritária a fim de equilibrar a quantidade de exemplos com o restante das classes. São apresentados no capítulo o *Oversampling* aleatório, bem como o SMOTE e algumas de suas variações.
- **Capítulo 4:** Apresentação da proposta: *Adversarial Oversampling - AO*. Consiste em um método para balanceamento de bases de dados envolvendo imagens, realizando a prática de *Oversampling*. Este capítulo descreverá a arquitetura do modelo, como suas camadas e etapas de treinamento. De forma resumida, o método utiliza uma AC-GAN, modificada para lidar com bases de dados desbalanceadas, com a técnica de *Minibatch Discrimination* e utilizando uma camada Ruído Gaussiano. O método proposto tem o treinamento dividido em duas etapas. A primeira etapa objetiva aprender a distribuição das geral das classes presentes na base de dados, com atenção especial para a classe minoritária. A segunda etapa é focada em aprender a gerar amostras similares às consideradas difíceis, pois elas são mais úteis para o classificador final. Por fim, é apresentada a etapa de filtragem das amostras geradas pelo método, para tentar evitar que algum tipo de ruído, imagem incoerente, passe a compor a base de dados. A etapa final consiste na inserção das imagens geradas para a classe minoritária, que passaram pela filtragem, na base de dados.
- **Capítulo 5:** Traz o protocolo experimental seguido para avaliar o método proposto. Os experimentos apresentados neste capítulo observam a qualidade das amostras geradas pelo AO em aspectos visuais, e também verificando o quão essas amostras contribuem com o aumento da taxa de acerto do classificador.

A primeira análise feita observa o quão as amostras geradas pelo método proposto contribuem para a precisão do classificador na classe minoritária. Uma outra análise observa o impacto que a etapa de filtragem tem na qualidade da geração das amostras, observando a precisão que o classificador consegue na classe minoritária com as amostras geradas com e sem a etapa de filtragem. Também é comparado o *Oversampling* feito através do AO, com os feitos pelos outros métodos tradicionais. A análise é feita observando o aspecto visual das amostras sintéticas geradas, além do quão elas contribuem para aprendizado dos classificadores finais. Tais classificadores foram determinados para serem redes neurais convolucionais difundidas no estado-da-arte: *ResNet*, *DenseNet* e *VGG16*. Além disso, são descritas as bases de dados e as medidas utilizadas para avaliação dos métodos.

- **Capítulo 6:** Apresenta os resultados dos experimentos realizados no capítulo anterior.
- **Capítulo 7:** Conclui o trabalho, resumizando o que foi feito e os resultados observados. Além de apontar possíveis trabalhos futuros para aperfeiçoamento do método proposto.

2 REFERENCIAL TEÓRICO

Esta seção traz o conteúdo da literatura que serviu de base para o desenvolvimento do presente trabalho. Primeiramente, a problemática envolvendo bases de dados desbalanceadas será descrita na Seção 2.1. Logo após, uma breve menção a uma possível técnica conhecida como *Undersampling*, na Seção 2.1.1. Porém sem entrar em detalhes, pois o objetivo do trabalho é a técnica de *Oversampling*, apresentada logo após, na Seção 2.1.2. Posteriormente, na Seção 2.2 são apresentadas as Redes Neurais Adversárias, bem como algumas suas variações. Por fim, são apresentadas técnicas, utilizadas para desenvolvimento do método proposto, úteis para melhorar o treinamento das Redes Adversárias e também para aperfeiçoar as amostras a serem geradas.

2.1 DESBALANCEAMENTO EM BASES DE DADOS

Em aprendizagem supervisionada, desbalanceamento de dados se refere ao problema em bases de dados na qual as classes não são representadas igualmente, ou seja, uma base é dita desbalanceada quando existem menos exemplos de algumas classes em relação às outras (BATISTA; PRATI; MONARD, 2004). Por exemplo, considerando um problema de classificação binária onde a base possui 100 exemplos, dos quais 80 são da classe A, considerada majoritária, e somente 20 da classe B, considerada minoritária. Esse é um problema de desbalanceamento da base de dados na proporção 80:20, ou 4:1.

Muitos classificadores tradicionais são sensíveis a esse tipo de problema, geralmente ignoram a classe minoritária, com menor representação, e tendem a se especializar somente na classe majoritária, com maior quantidade de exemplos. Considerando que esse classificador seja um Rede Neural, por exemplo, durante o treinamento, as amostras pertencentes à classe minoritária serão apresentadas poucas vezes para o ajuste dos pesos sinápticos. Logo, a superfície de decisão será construída priorizando a classe que foi mais vezes apresentada, no caso, a classe majoritária.

Para reduzir o problema de desbalanceamento pode-se simplesmente coletar mais exemplos da classe minoritária. Porém, dependendo do domínio da base de dados, não é uma tarefa simples. As vezes a classe minoritária é considerada uma classe rara, dificultando a coleta de exemplos pertencentes à mesma. Mesmo que não seja tão rara, ainda pode exigir um trabalho considerável, pois é necessário que a coleta/adição desses exemplos siga a mesma estratégia e metodologia utilizada

na construção original da base. Isto é, com os mesmos equipamentos, sensores, técnicas, protocolos, etc.

Tendo em vista esse problema, algumas abordagens foram desenvolvidas para permitir que as bases desbalanceadas pudessem ser aproveitadas da melhor forma para a realização do aprendizado supervisionado. Como já mencionado no Capítulo 1, essas abordagens são agrupadas em três (JOHNSON; KHOSHGOFTAAR, 2019), (KRAWCZYK, 2016):

- No nível de dados: Trata a base de dados realizando alguma re-amostragem a fim de equilibrar a proporção das classes. Nas próximas seções serão dados mais detalhes;
- No nível de algoritmo: Adapta o modelo de aprendizagem para lidar com o desequilíbrio das classes durante o treinamento, atribuindo um custo maior para uma classe menos representada, aumentando sua importância durante o processo de aprendizagem. Com essa adaptação, a classe minoritária ganha mais importância e o viés sobre a classe majoritária é reduzido (LIU; LIU, 2012).
- Híbrida: Combinação dos métodos a nível de dados e a nível de algoritmo. Como por exemplo, realizar uma re-amostragem através de uma abordagem no nível de dados e, posteriormente, ajustar o modelo adicionando um custo maior para a classe originalmente minoritária.

A nível de dados, existem os métodos que aumentam a quantidade de amostras minoritárias, chamados de *Oversampling*, e os métodos que diminuem a quantidade de amostras das classes majoritárias, chamados de *Undersampling*. Alguns pontos relacionados a esses métodos serão descritos a seguir.

2.1.1 Undersampling

Uma forma de equilibrar a quantidade de amostras por classe é remover exemplos da classe majoritária, essa forma é conhecida como *Undersampling*. O método mais simples para se realizar *Undersampling* é através do *Random Undersampling - RUS*, no qual as amostras da classe majoritária a serem removidas são selecionadas aleatoriamente. Essa remoção ocorre para que a classe majoritária possua a mesma quantidade de amostras da classe minoritária. Porém, para que o *Undersampling* seja seguro, tem que seguir alguns critérios para determinar quais amostras permanecerão e quais serão descartadas, caso contrário informações importantes sobre uma a classe podem ser perdidas. Pois quanto mais informação, corretamente

ligada à classe, for passada ao classificador, melhor ele será em estimar a distribuição dessa classe e gerar uma superfície de decisão adequada (JOHNSON; KHOSHGOFTAAR, 2019), (KRAWCZYK, 2016).

2.1.2 *Oversampling*

Uma outra forma de equilibrar a proporção de exemplos por classe é realizando a inserção de novas amostras pertencentes à classe minoritária, para que a mesma fique balanceada com as demais, essa forma é conhecida como *Oversampling*. Tal técnica mantém todas as amostras da classe majoritária e não perde informações sobre a mesma. Com isso, mais informação é disponibilizada para o treinamento do modelo, o que é um fator muito importante, principalmente se o modelo de classificação for uma rede neural profunda. O método mais simples de se realizar *Oversampling* é através do *Random Oversampling - ROS*, que seleciona aleatoriamente amostras da classe minoritária e as repete na base de dados, até que a proporção de classes fique equilibrada. Entretanto, a técnica de *Oversampling* tem que ser aplicada com precaução, pois essas novas amostras têm que seguir a distribuição da classe a qual elas estão sendo inseridas. Um problema que também pode ocorrer é o chamado "ruído de classe", quando uma nova amostra inserida pertence à distribuição de outra classe do problema. A inserção de uma amostra que não pertença à distribuição da classe pode prejudicar ainda mais o aprendizado dessa classe por parte do modelo a ser treinado (JOHNSON; KHOSHGOFTAAR, 2019), (KRAWCZYK, 2016).

2.2 *GENERATIVE ADVERSARIAL NETWORKS - GANS*

Ainda se tratando de geração de amostras sintéticas, porém sem, necessariamente, ter foco em *Oversampling*, foram apresentadas as Redes Neurais Adversárias, GANs (GOODFELLOW et al., 2014). Esse método vem sendo objeto de pesquisa na tarefa geração de amostras sintéticas em vários domínios (GECER et al., 2018), (KARRAS et al., 2017), (JIN et al., 2017), (ZHU et al., 2017), (ZHANG et al., 2016), (REED et al., 2016), (MA et al., 2017), (TAIGMAN; POLYAK; WOLF, 2016), (LIU; TUZEL, 2016).

As GANs são compostas por duas redes neurais, que competem entre si numa espécie de jogo entre um "falsificador" e um "avaliador". O primeiro, é representado pela Rede Geradora **G**, que, dado um conjunto de treinamento **X** e um vetor de aleatório **z**, tenta gerar amostras sintéticas similares às originais. O "avaliador" é representado pela Rede Discriminadora **D**, que é responsável por aprender a discriminar as amostras sintéticas geradas por **G**, das amostras reais presentes no conjunto de treinamento **X**. Dessa forma, dado uma amostra sintética \mathbf{x}_s , gerada por **G**, **D** aprende a classificar essa amostra em *real* (pertencente à **X** ou *falsa* (ge-

rada por **G**. Um exemplo do que ocorre com **D** e **G** pode ser visto na Figura 1. A arquitetura final das GANs, com **D** e **G** inseridos, pode ser visto na Figura 2.

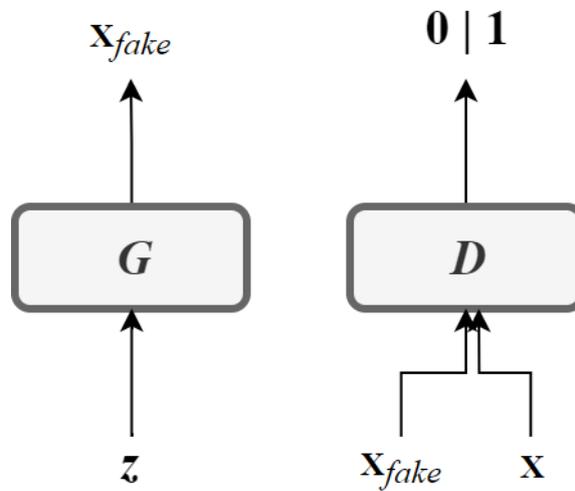


Figura 1 – Ilustração do funcionamento do Gerador **G**, o qual recebe um ruído z e gera uma amostra X_{fake} , e do Discriminador **D**, recebe X_{fake} e X e as classifica em real ou falso (0 ou 1)

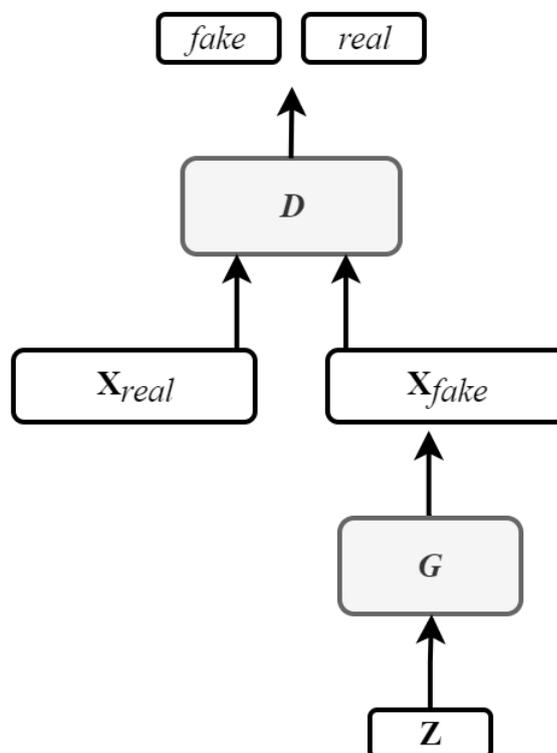


Figura 2 – Arquitetura geral das GANs

O objetivo de **G** gerar dados sintéticos para confundir **D**, cujo papel é ser capaz de distinguir entre dados verdadeiros e dados gerados por **G**. Dessa forma, durante o treinamento de **G** tem-se o objetivo de maximizar o erro, enquanto objetiva-se

minimizá-lo para \mathbf{D} . Isso resulta na fórmula presente na Equação 2.1. Os dois modelos, \mathbf{D} e \mathbf{G} são treinados juntos em um jogo de soma zero onde, se um ganha, o outro perde. Esse treinamento ocorre até que o modelo discriminador seja enganado na mesma proporção que acerte, o que significa que o modelo gerador está gerando exemplos próximos da realidade e que o modelo discriminador está conseguindo avaliar esses exemplos.

$$\max_{\mathbf{G}} \left(\min_{\mathbf{D}} E(\mathbf{G}, \mathbf{D}) \right) \quad (2.1)$$

2.2.1 Conditional Generative Adversarial Networks - CGAN

Dado um conjunto de treinamento \mathbf{X} , contendo várias classes, as GANs tradicionais são capazes de gerar amostras sintéticas pertencentes a todas as classes. Entretanto, não é possível controlar quando cada classe é gerada, pois o treinamento ocorre de forma não-supervisionada. Com isso, as GANs tradicionais guardam informação da distribuição das classes de forma conjunta e sem distinção entre elas. Para tornar possível a geração de amostras condicionadas, as CGANs foram propostas (MIRZA; OSINDERO, 2014). Nesse tipo de GANs o modelo generativo \mathbf{G} é treinado para gerar exemplos condicionados por alguma entrada condicional c . Considerando que \mathbf{X} é uma base de dígitos, essa entrada pode ser o rótulo de uma classe, dígito 1, por exemplo. Além de \mathbf{G} , o discriminador \mathbf{D} também é condicionado. Ele recebe como entrada uma imagem real ou falsa e a entrada condicional, c . No caso de uma entrada condicional sendo o rótulo de uma classe, o discriminador esperaria que a entrada fosse dessa classe, forçando o gerador a gerar exemplos da classe c . Um exemplo do que ocorre de novo com \mathbf{D} e \mathbf{G} pode ser visto na Figura 4. A arquitetura final das CGANs, com \mathbf{D} e \mathbf{G} inseridos e recebendo suas entradas condicionais, c , pode ser visto na Figura 3.

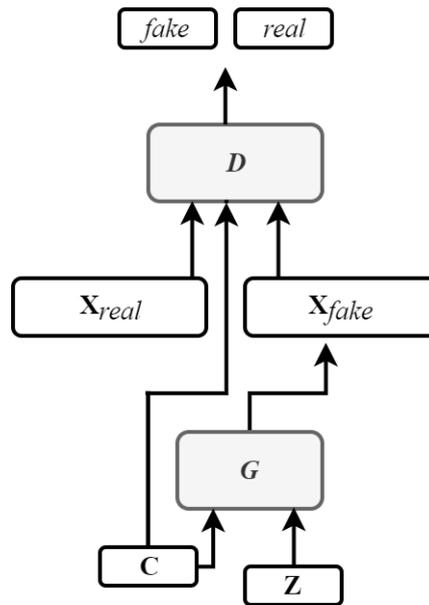


Figura 3 – Arquitetura geral da CGANs

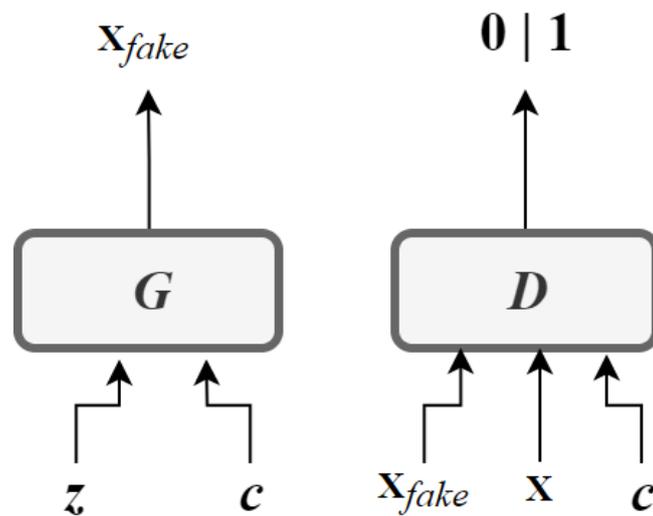


Figura 4 – Ilustração do funcionamento do Gerador \mathbf{G} , o qual recebe um ruído \mathbf{z} , uma informação condicional c e gera uma amostra \mathbf{X}_{fake} condicionada à c , e do Discriminador \mathbf{D} , que recebe \mathbf{X}_{fake} , \mathbf{X} e c e classifica-os em real ou falso (0 ou 1) de forma condicionada à c

2.3 AUXILIARY CLASSIFIER GANS - AC-GAN

A *Auxiliary Classifier GANs* (AC-GAN) (ODENA; OLAH; SHLENS, 2016) é uma variante das GANs, mais especificamente, da CGANs, a qual já foi devidamente descrita no final do Capítulo 2. De forma diferente das GANs tradicionais, a AC-GAN, assim como a CGANs, recebe como entradas um ruído \mathbf{z} , que servirá como base para a geração da amostra sintética, e uma classe $c \sim p_c$. Essa classe condiciona o Gerador a gerar amostra que sigam as mesmas características dessa classe indicada $\mathbf{X}_{fake} = \mathbf{G}(c, \mathbf{z})$. Porém, difere da CGANs, pois o Discriminador \mathbf{D} da AC-GAN recebe

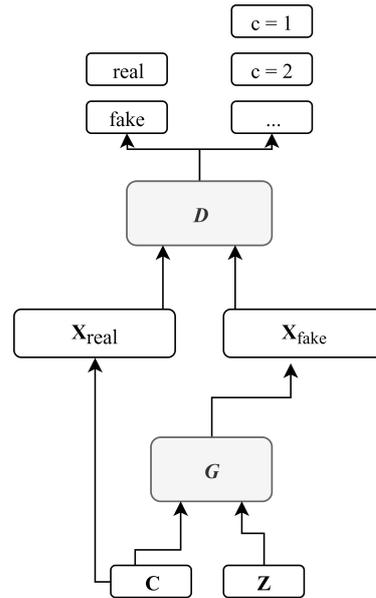


Figura 5 – Arquitetura da AC-GAN, onde **D** é o Discriminador, **G** o Gerador e c e z são a classe e o ruído, respectivamente.

somente a imagem sintética como entrada. Dessa forma, o papel do Discriminador é prever se a imagem recebida é real ou não, além da classe que ela pertence. O Discriminador resultará numa distribuição de probabilidade dos dados de treinamento \mathbf{X} e uma distribuição de probabilidade das classes C , Equação 2.2.

$$P(\mathbf{S}|\mathbf{X}), P(C|\mathbf{X}) = \mathbf{D}(\mathbf{X}) \quad (2.2)$$

As AC-GAN fornecem isso a partir da seguinte função objetivo:

$$L_S = E[\log P(\mathbf{S} = \text{real}|\mathbf{X}_{\text{real}})] + E[\log P(\mathbf{S} = \text{fake}|\mathbf{X}_{\text{fake}})] \quad (2.3)$$

$$L_C = E[\log P(C = c|\mathbf{X}_{\text{real}})] + E[\log P(C = c|\mathbf{X}_{\text{fake}})] \quad (2.4)$$

A partir do treinamento utilizando as funções acima, o Discriminador é treinado para maximizar a probabilidade de classificar corretamente imagens reais e falsas, L_S , e prever corretamente a classe, L_C , da respectiva imagem, logo $(L_S + L_C)$. Já o Gerador é treinado para "enganar" o Discriminador, maximizando $L_C - L_S$.

Segundo (ODENA; OLAH; SHLENS, 2016), as AC-GAN possuem um treinamento mais estável e o Gerador é capaz de gerar imagens com uma alta resolução. Sua arquitetura é sumarizada na Figura 5.

2.4 CAMADA DE RUÍDO GAUSSIANO

Quando redes neurais, principalmente as profundas, são utilizadas para treinamento a partir de bases de dados pequenas, um problema comum é o chamado

overfitting, no qual a rede memoriza o conjunto de dados de treinamento, perde sua capacidade de generalização. Em vez de aprender um mapeamento do conjunto de dados, como um todo, o modelo aprende os exemplos de entradas específicas e suas saídas associadas. Como já falado, isso irá resultar em um modelo com bom desempenho no conjunto de dados de treinamento, mas um desempenho inaceitável em dados ainda não apresentados, como os de validação ou até mesmo os de teste. Um pequeno conjunto de dados oferece menos oportunidade de aprender a real distribuição do problema, facilitando a ocorrência de *overfitting*.

Para diminuir dos problemas citados acima, pode ser adicionado um ruído z aleatório à rede, com isso melhorar seu erro generalização (NEELAKANTAN et al., 2015), (SØNDERBY et al., 2016). Mesmo soando de forma pouco intuitiva, pois pode-se esperar que o ruído prejudique o desempenho do modelo durante o treinamento, na prática, foi demonstrado que o treinamento com ruído pode realmente levar a melhorias à generalização da rede. A adição de ruído implica em fazer a rede menos capaz de memorizar os dados de treinamento, pois eles estão mudando a todo momento, resultando em uma rede mais robusta e apresentando maior capacidade de generalização, sem evitar que a rede aprenda a distribuição dos dados. Dessa forma, a adição de uma camada de ruído gaussiano de média zero e desvio padrão 1, $z \sim N(\mu = 0, \sigma^2 = 1)$, pode ser adicionada na rede.

2.5 MINIBATCH DISCRIMINATION

Como mostrado na Seção 2.2, as GANs tradicionalmente são compostas por uma rede discriminadora \mathbf{D} e uma rede geradora \mathbf{G} . Durante o treinamento, \mathbf{G} tem que conseguir enganar \mathbf{D} , gerando amostras tão reais quanto as originais. Um problema comum é o colapso da rede, que significa que \mathbf{G} gera sempre a mesma amostra, pois essa amostra é considerada real por \mathbf{D} . Esse é um problema grave, pois é esperado que as GANs sejam capazes de estimar a distribuição geral dos dados reais, para assim poder gerar amostras diversas, não somente uma mesma amostra que pertença a essa distribuição.

Para resolver esse problema, o conceito chamado *Minibatch-Discrimination* (SALIMANS et al., 2016) foi apresentado. A ideia por trás consiste em fazer o Discriminador considerar as informações presentes em um *batch* de amostras, em vez de considerar apenas as informações fornecidas por uma única entrada. Dessa forma, o colapso da rede fica perceptível, pois o Discriminador conseguirá perceber a informação global de similaridade entre as amostras e, sempre que todas as amostras de um lote estiverem muito próximas umas das outras, elas serão consideradas falsas, forçando o Gerador a fornecer amostras mais diversas.

Para verificar o quão próximas estão as imagens que compõe um determinado *batch*, a solução apresentada foi calcular a norma L_1 entre as características relacionadas às imagens de entrada. Quanto menor a norma L_1 , mais semelhantes são as entradas de dados. Essa nova informação é concatenada com a entrada original e alimentada na próxima camada do Discriminador, de modo que o gerador possa alterar seu comportamento de acordo com essa similaridade. Com isso, se a similaridade das imagens geradas aumentar, significa que o colapso do modelo está próximo, e o Discriminador pode passar a negar essas imagens geradas, forçando o Gerador a reajustar seus parâmetros para aumentar a diversidade na geração de amostras. Segundo os autores, com a técnica de *Minibatch-Discrimination* é permitido gerar amostras, visualmente atraente, muito rapidamente.

2.6 FILTRAGEM DE DADOS SINTÉTICOS GERADOS POR GANS

As amostras sintéticas geradas por GANs já se mostraram úteis para aumentar a base de dados no intuito de melhorar a acurácia do classificador, como visto ao final do Capítulo 1. Entretanto, a geração pode nem sempre fornecer dados úteis, com aspectos reais, ou até diversos. Para lidar com isso, foi proposta uma abordagem para realizar a amostragem dos dados sintéticos gerados (BHATTARAI et al., 2019). São três métodos apresentados: *Class conditional probability*, *Realism conditional probability* e utilizando *Reinforcement Learning*. Os dois primeiros métodos irão compor a proposta do presente trabalho, logo serão descritos brevemente à seguir.

2.6.1 *Class conditional probability*

A primeira proposta, é utilizando a probabilidade como uma espécie de *score* de confiabilidade da classe. Esse *score* é dado através da probabilidade condicional de classe calculada através da Equação 4.8.

$$score(\mathbf{x}_s) = P(y_c | \mathbf{x}_s, \theta_c). \quad (2.5)$$

Onde, \mathbf{x}_s é a amostra sintética gerada pela GANs, y_c a classe alvo e θ_c é um classificador previamente treinado, utilizando somente o conjunto real. Quanto maior a confiabilidade da amostra \mathbf{x}_s pertencer a classe em questão, maior as chances dessa amostra incorporar o *dataset*.

2.6.2 *Realism conditional probability*

A segunda proposta tem o objetivo em selecionar as imagens sintéticas de acordo com o realismo que possuem. Para isso, o *score* de realismo é dado através da

própria GANs utilizada, mais especificamente do discriminador **D**. Quanto maior o *score* de realismo dado por **D**, recebendo \mathbf{x}_s , maior é a chance de que a amostra seja utilizada para compor o *dataset*.

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados alguns trabalhos que estão relacionados com a problemática da atual pesquisa, desbalanceamento de classes. Os trabalhos apresentados neste capítulo se concentram na prática de *Oversampling*, ou seja, geram amostras para a classe minoritária até que o problema de desbalanceamento seja resolvido. Primeiramente, é apresentado o *Random Oversampling*, logo após, o SMOTE e suas variações: ADASYN, Borderline-SMOTE, SVMSMOTE e, por fim, o KMeans-SMOTE. Esses métodos serão descritos nas seções a seguir.

3.1 RANDOM OVERSAMPLING - ROS

Random Oversampling (ROS) é considerada a forma mais simples de se realizar *Oversampling*. O método consiste em escolher amostras da classe minoritária, aleatoriamente, e as repetir. Esse método não considera nenhuma informação da distribuição. Mesmo claramente sendo uma forma simples, ainda costuma ser utilizado para lidar com conjuntos de dados desbalanceados.

3.2 SYNTHETIC MINORITY OVERSAMPLING - SMOTE

O método mais tradicional para realização de *Oversampling* é o *Synthetic Minority Over-sampling Technique* (SMOTE) (CHAWLA et al., 2002), que obteve um grande sucesso na tarefa de diminuir os efeitos do desbalanceamento de classes em bases de dados. Para gerar novas amostras, o SMOTE seleciona aleatoriamente uma amostra pertencente à classe minoritária e considera os k vizinhos dessa amostra. A partir dessa vizinhança, uma nova amostra é gerada realizando uma interpolação desses pontos vizinhos. Para isso, são desenhadas linhas entre as amostras que compõem a vizinhança, e sobre essas linhas são geradas amostras (pontos) sintéticas pertencentes à classe minoritária. Um exemplo da execução do SMOTE pode ser visto nas imagens A e B, da Figura 6. A imagem A traz um exemplo ilustrativo de uma base de dados onde o símbolo \bullet representa a classe minoritária e o símbolo $+$ representa a classe majoritária. Na imagem, B pode-se ver as amostras geradas, em verde, sobre as linhas que ligam a vizinhança das amostras selecionadas, em vermelho. A geração se dá pela Equação 3.1, onde \mathbf{s}_i é amostra gerada a partir de \mathbf{x}_i e seu vizinho mais próximo \mathbf{x}_{ki} . Para adicionar uma aleatoriedade, é utilizado um valor λ entre 0 e 1.

$$\mathbf{s}_i = \mathbf{x}_i + (\mathbf{x}_{ki} - \mathbf{x}_i)\lambda; \quad (3.1)$$

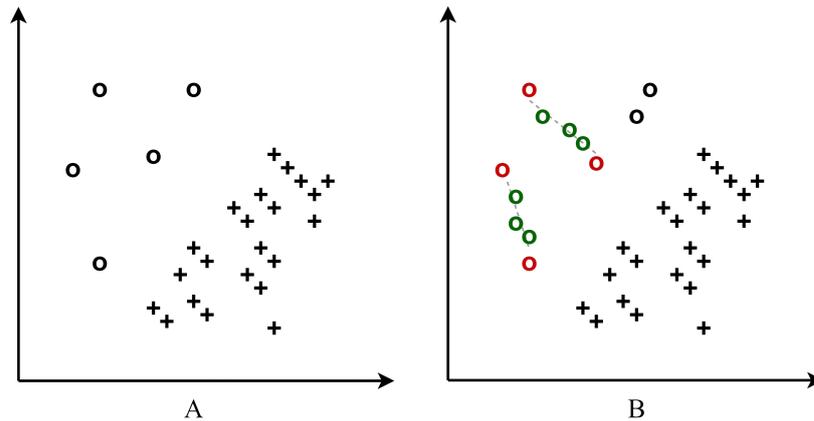


Figura 6 – Amostras sintéticas geradas pelo SMOTE, para a classe minoritária representada por \bullet . Em verde as amostras sintéticas e em vermelho as amostras reais, utilizadas na geração.

A principal vantagem do SMOTE em comparação com o ROS é que, no o processo de criação de amostras sintéticas, o classificador não se depara com amostras repetidas. Por isso, é menos provável que o classificador se ajuste demais durante o treinamento.

3.3 ADAPTIVE SYNTHETIC - ADASYN

Diferente do método SMOTE, o *Adaptive Synthetic* (ADASYN) (Haibo He et al., 2008) o considera a dificuldade das amostras da classe minoritária para determinar quantas amostras serão geradas a partir de cada amostra da classe minoritária. A ideia por trás dessa condição é gerar amostras sintéticas mais úteis para o classificador. Isso acontece porque a quantidade de exemplos que uma amostra minoritária irá gerar, será proporcional à quantidade de vizinhos da classe majoritária dessa amostra. Isso faz com que mais amostras sejam geradas próximas à região de fronteira da classe minoritária. Para execução do ADASYN, os seguintes passos são seguidos:

- Considerando m_s e m_l as quantidades de amostras da classes minoritárias e majoritárias, respectivamente, obtém-se o grau de desbalanceamento d da base de dados

$$d = \frac{m_s}{m_l}; \quad (3.2)$$

- Com β sendo o nível de balanceamento desejado, calcula-se a quantidade G de amostras sintéticas à serem geradas para a classe minoritária

$$G = (m_l - m_s)\beta; \quad (3.3)$$

- Para cada amostra da classe minoritária encontram-se os k vizinhos mais próximos e observa-se a dominância r_i da classe majoritária, onde Δ_i indica a

quantidade de vizinhos, da amostra i , que pertencem a classe majoritária

$$r_i = \frac{\Delta_i}{k}; \quad (3.4)$$

- Normaliza-se os valores r_i

$$\hat{r}_i = \frac{r_i}{\sum r_i}; \quad (3.5)$$

- Para cada vizinho, calcula-se o número de amostras sintéticas a serem geradas

$$\hat{G}_i = G\hat{r}_i; \quad (3.6)$$

- Finalmente, é gerado a amostra sintética \mathbf{s}_i utilizando as amostras minoritárias \mathbf{x}_i e \mathbf{x}_{ki} pertencentes a mesma vizinhança. A geração é dada pela equação abaixo

$$\mathbf{s}_i = \mathbf{x}_i + (\mathbf{x}_{ki} - \mathbf{x}_i)\lambda; \quad (3.7)$$

Onde λ é um número aleatório entre 0 e 1.

3.4 BORDERLINE-SMOTE

O próximo método para realização de *Oversampling* é o *Borderline SMOTE* (BSMOTE) (HAN; WANG; MAO, 2005), o qual faz uma busca na base de dados para encontrar amostras, da classe minoritária, consideradas difíceis. A partir dessas amostras selecionadas, novas amostras são geradas e o *Oversampling* é efetuado na classe minoritária. O procedimento ocorre da seguinte forma:

- Para cada amostra \mathbf{x}_m da classe minoritária, são calculados os k vizinhos mais próximos considerando o todo o conjunto de treinamento \mathbf{X} ;
- Se todos os k vizinhos mais próximos forem da classe majoritária, ou de outra classe além da minoritária, então \mathbf{x}_m é considerada um ruído e é desconsiderado do procedimento;
- Se \mathbf{x}_m possui todos os vizinhos da classe minoritária, então \mathbf{x}_m é considerada uma amostra fácil de ser classificada. Pois teoricamente não irá contribuir consideravelmente para o classificador;
- Caso os dois itens mencionados anteriormente não sejam atendidos, então \mathbf{x}_m é considerada uma amostra importante, e é inserida no conjunto de amostras consideradas importantes. As amostras pertencentes à esse conjunto servirão como base para geração de amostras sintéticas e, conseqüentemente, para realização do *Oversampling*;

- Finalmente, o conjunto de amostras consideradas importantes é utilizado pelo SMOTE tradicional para geração de amostras sintéticas.

3.5 SVMSMOTE

O próximo método de lidar com a tarefa de *Oversampling* foi apresentado o *SVM-SMOTE* (SVMSMOTE) (NGUYEN; COOPER; KAMEI, 2011). Tal método realiza a geração de amostras sintéticas próximas a região ótima de decisão. O SVMSMOTE concentra-se em gerar amostras próximas à região de borda da classe, utilizando apenas amostras que compõe à região ótima de decisão, não sendo necessária a utilização todas as amostras da classe minoritária. A região ótima de decisão é obtida através do treinamento de *SVMs* (Hearst et al., 1998) na base de dados original. Diferente dos métodos citados anteriormente, o SVMSMOTE gera amostras sintéticas através de extrapolação ou interpolação. A escolha do método de geração depende da quantidade de amostras majoritárias em volta da respectiva amostra minoritária, funcionando da seguinte forma:

- **Extrapolação:** Ocorre quando a quantidade de amostras pertencentes a classe majoritária for menor que a metade da quantidade total de vizinhos. Dessa forma, a extrapolação será feita e isso irá expandir a área da classe minoritária. Isso pode ser visto no exemplo no lado esquerdo da Figura 7;
- **Interpolação:** É realizada se a quantidade de amostras pertencentes a classe majoritária for maior ou igual a metade quantidade total de vizinhos. Dessa forma, similarmente ao SMOTE tradicional, a borda da classe minoritária será consolidada. Isso pode ser visto no exemplo no lado direito da Figura 7.

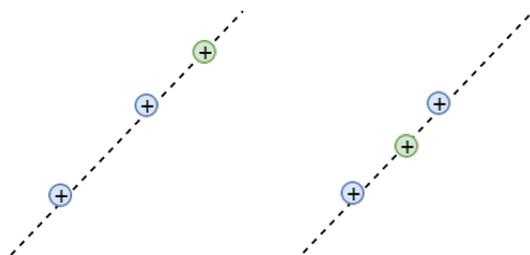


Figura 7 – Exemplo da aplicação do SVMSMOTE. No lado esquerdo ocorre a extrapolação, onde a amostra sintética (verde) extrapola a região entre os exemplos minoritários (azul). Já no lado direito, ocorre a interpolação, onde a amostra sintética (verde) é gerada entre os exemplos minoritários (azul).

3.6 KMEANS-SMOTE

O próximo método para realização de *Oversampling* é o *K-Means SMOTE* (*KS-MOTE*) (LAST; DOUZAS; BAÇÃO, 2017), que é mais uma variação da tradicional técnica *SMOTE*. Como o próprio nome sugere, o método aplica o *K-Means* (Lloyd, 1982) para geração de *clusters* a partir da base de dados. A elaboração desses *clusters* tem o objetivo de dividir as amostras em k grupos. A partir disso, é feita uma filtragem dos *clusters*, mantendo apenas os que possuem pelo menos 50% de amostras minoritárias. Por fim, os *clusters* filtrados são utilizados como base para geração de amostras sintéticas para a classe minoritária, através do método *SMOTE*.

A seguir, na Tabela 1, é apresentada uma sumarização dos métodos de *Oversampling* apresentados, a qual contém as diferenças na forma de escolha do exemplos da classe minoritária que participarão do *Oversampling* e a forma de geração das amostras artificiais.

Tabela 1 – Sumarização dos métodos de *Oversampling* apresentados

Método	Escolha do exemplo que participará do <i>Oversampling</i>	Geração
ROS	Escolhe aleatoriamente amostras e as repete.	–
SMOTE	Escolhe aleatoriamente amostras e gera amostras artificiais.	Interpolação
ADASYN	Determina quantas amostras artificiais cada amostra minoritária irá gerar através da interpolação.	Interpolação
BSMOTE	Escolhe amostras difíceis para geração das amostras artificiais.	Interpolação
SVMSMOTE	Escolhe amostras próximas a região ótima de decisão.	Extrapolação e Interpolação
KSMOTE	Gera clusters da base de dados e escolhe clusters que possuem pelo menos 50% amostras minoritárias para gerar amostras artificiais.	Interpolação

4 MODELO PROPOSTO

Nas próximas seções está de forma detalhada a principal contribuição do presente trabalho: o *Adversarial Oversampling* - AO. Utilizando como base o conteúdo já apresentado nos capítulos anteriores, o método proposto será formulado e apresentado detalhadamente, em todas as suas etapas. Primeiramente, é descrita a arquitetura das redes adversárias utilizadas como base do AO, incluindo as adaptações utilizadas para lidar com bases de dados desbalanceadas. Além disso, é apresentada o fluxo de treinamento e a filtragem proposta.

4.1 ADVERSARIAL OVERSAMPLING - AO

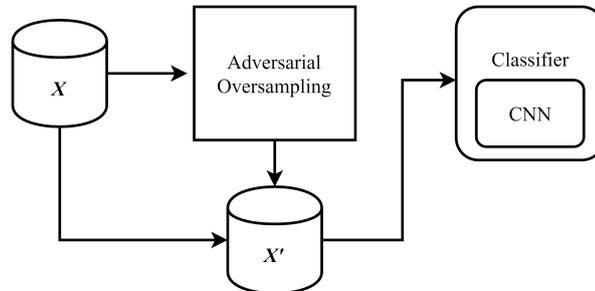
Tendo em vista a problemática, trazida no fim do Capítulo 1, e o potencial das GANs para geração de amostras sintéticas, o presente trabalho traz a proposta chamada *Adversarial Oversampling* - AO. Como sugere o nome, consiste em um *pipeline* para *Oversampling*, em bases de dados envolvendo imagens, utilizando o aprendizado adversário trazido nas GANs. O processo adotado pelo AO inicia-se utilizando uma arquitetura modificada das AC-GAN, para melhor adaptação ao desbalanceamento das bases de dados. Essa modificação consiste na inserção das camadas: Ruído Gaussiano, que evita o *overfitting* causado pela repetição das amostras da classe minoritária; e Minibatch-Discrimination, que força a diversidade das amostras geradas. Além da modificação na arquitetura, o método possui um pipeline para treinamento com foco em geração de amostras mais úteis para o classificador. Nas próximas seções, serão exibidas detalhadamente as camadas e etapas definidas para o AO na realização do *Oversampling*.

4.1.1 Utilizando o *Adversarial Oversampling*

Como retratado, o *Adversarial Oversampling* consiste em um método para balanceamento de bases envolvendo imagens. Para sua utilização, é necessário que a base de dados seja representada por imagens ou ao menos tenha uma representação espacial 2D (ou 3D), ou seja, a base seja composta por exemplos com as características possuindo relação espacial, como uma imagem, por exemplo. Pois a base do método de geração é uma AC-GAN com redes convolucionais, próprias para extrair e tirar proveito, durante o treinamento, das relações espaciais que compõem os exemplos da base. Na Figura 8 podemos ver o esquema de uso do AO, com o qual é realizado o *Oversampling* sobre a base desbalanceada \mathbf{X} , gerando \mathbf{X}' , a qual será

utilizada para treinar o classificador, no caso, uma *Convolutional Neural Networks* (CNN).

Figura 8 – Fluxo do uso *Adversarial Oversampling* para balancear o a base de dados \mathbf{X} .

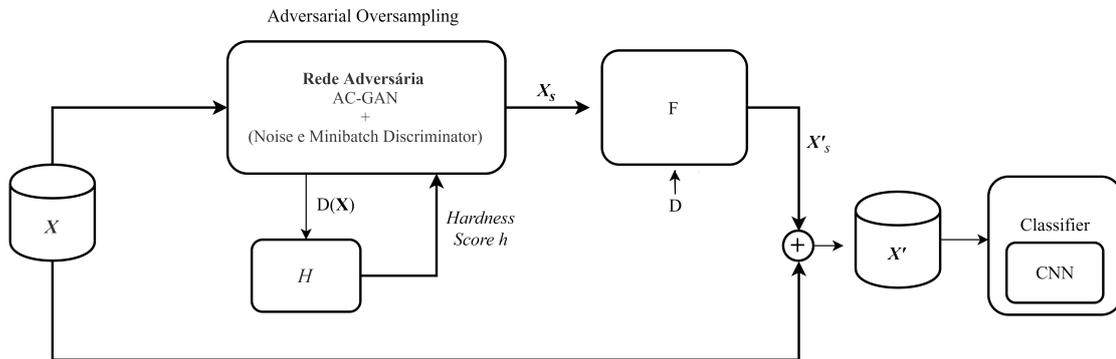


4.1.2 Arquitetura Geral do Modelo

A arquitetura do AO consiste basicamente em três em módulos: Rede Adversária, que é responsável pela base do modelo, aprendendo a distribuição dos dados e gerando amostras sintéticas que sigam a mesma distribuição e possam contribuir para o classificador final; O módulo de entropia, representado por H , que é responsável por contribuir na segunda parte do treinamento da Rede Adversária, fazendo a mesma focar em gerar amostras difíceis; Por fim, o terceiro módulo é responsável por realizar a filtragem das amostras sintéticas geradas pelo AO. Uma forma sumariada da arquitetura geral do AO, com seus módulos, pode ser vista na Figura 9. Primeiramente, a base de dados \mathbf{X} , originalmente desbalanceada, é utilizada pela Rede Adversária para que seja possível aprender a distribuição das classes. Utilizando o discriminador \mathbf{D} , H determina quais amostras deverão ser apresentadas mais frequentemente no treinamento. Após a etapa de treinamento, as amostras geradas por são filtradas em F e passam a compor o novo conjunto de treinamento balanceado, representado por \mathbf{X}' . Por fim, \mathbf{X}' é utilizado para aprendizagem de um classificador em questão, no caso, uma CNN.

4.1.3 Arquitetura das Redes Adversárias

A rede adversária que serve de base para o modelo proposto é a AC-GAN (ODENA; OLAH; SHLENS, 2016). Essa rede foi escolhida pois é uma variante tradicional das GANs e possibilita o uso das informações de distribuição das classes durante o treinamento. Além disso, os autores mostraram que, com uso da AC-GAN, foi possível observar uma maior estabilidade no treinamento, além da possibilidade de geração de imagens com uma alta resolução. Como geralmente ocorre na maioria das

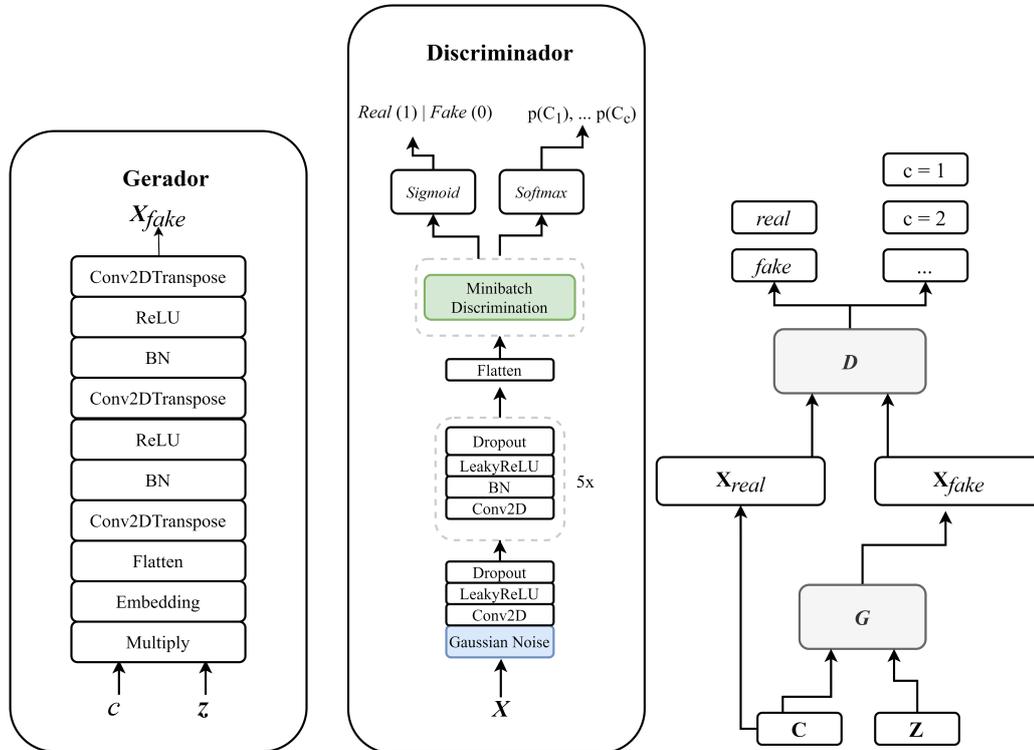
Figura 9 – Arquitetura geral do *Adversarial Oversampling*.

GANs, a arquitetura consiste em duas redes: Gerador (**G**) e Discriminador (**D**). A arquitetura utilizada possui componentes escolhidos de acordo com os recomendados (SALIMANS et al., 2016). Tanto o **G**, quanto o **D**, são compostos por camadas convolucionais, pois são utilizados sobre imagens, com dados possuindo características espaciais, logo as convoluções são mais recomendadas. A arquitetura das Redes Adversárias utilizadas no modelo pode ser vista na Figura 10. Do lado esquerdo, tem-se o Gerador **G**, no centro o Discriminador **D** e no lado direito, o esquema de treinamento, que não difere do tradicional envolvendo AC-GAN. Ainda no lado direito, pode-se notar o **D** com as saídas da esquerda, responsável por determinar se a imagem é real ou falsa, e da direita, com a probabilidade da imagem pertencer a uma das c classe do problema.

4.1.3.1 Rede Geradora (**G**)

Ilustrado no lado esquerdo da Figura 10, **G** é responsável por aprender a distribuição dos dados e gerar amostras sintéticas pertencentes à mesma. Para isso, recebe a classe c , condicionando a geração à distribuição dessa classe, e $\mathbf{z} \sim N(\mu = 0, \sigma = 0.5)$, o ruído aleatório a partir do qual a amostra sintética será gerada. No início há uma camada de multiplicação, na qual é feito o produto *Hadamard* entre a parte da rede que aprende a distribuição dos dados, utilizando \mathbf{z} , com a parte responsável pelo rótulo da classe, c . **G** é composto por um conjunto de blocos convolucionais, os quais possuem uma camada convolucional, a função de ativação *Rectified Linear Unit* (ReLU) (NAIR; HINTON, 2010), Equação 4.1, e uma camada de *Batch Normalization* (BN) (IOFFE; SZEGEDY, 2015). Essas camadas são comumente utilizadas em redes neurais convolucionais. Como saída, **G** retorna a imagens falsas (\mathbf{X}_{fake}). Ao longo do treinamento, essas imagens vão, cada vez mais, possuindo a aparência

Figura 10 – Arquitetura das Redes Adversárias utilizadas como base do *Adversarial Oversampling*. No lado esquerdo é exibido a estrutura do Gerador **G** e do Discriminador **D**, o qual contém as camadas de ruído gaussiano e *Minibatch-Discrimination*. No lado direito, está o fluxo de treinamento, seguindo o padrão das AC-GAN



das imagens reais.

$$f(x) = ReLU(x) = \max(0, x) \quad (4.1)$$

4.1.3.2 Rede Discriminadora (**D**)

No centro da Figura 10 pode-se observar a composição da rede **D**, a qual também consiste em uma rede convolucional. É composta por blocos convolucionais, formados por uma camada convolucional, *Batch Normalization* (BN) e, como função de ativação, a *Leaky Rectified Linear Unit* (Leaky-ReLU) (MAAS; HANNUN; NG, 2013), Equação 4.2. Ainda em **D**, percebe-se uma camada de *Dropout* (SRIVASTAVA et al., 2014) ao fim de cada bloco convolucional. Esses blocos convolucionais são repetidos 5 vezes, concatenando cada saída com a entrada do próximo bloco convolucional. Por fim, a rede **D** possui duas saídas: O bloco à esquerda, possuindo a função de ativação *Sigmoid*, Equação 4.3, que retorna se a imagem gerada é real(1) ou falsa(0); O bloco a direita utiliza a função 4.4 para retornar as probabilidades das classes a qual essa imagem gerada pertence ($p(C_1), \dots, p(C_c)$), onde c é a quantidade

de classes do problema.

$$f(x) = \text{LeakyRelU}(x) = \max(0.01x, x) \quad (4.2)$$

$$f(x) = \text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (4.3)$$

$$f(y_i) = \text{Softmax}(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (4.4)$$

O presente trabalho também incorpora em **D** duas técnicas, facilitando seu aprendizado em bases desbalanceadas: *Minibatch-Discrimination* e o ruído gaussiano (*Gaussian Noise*). Essas técnicas foram apresentadas nas Seções 2.5 e 2.4, respectivamente. A motivação para adoção dessas técnicas no presente trabalho será descrita a seguir.

- **Gaussian Noise:** Um problema comum envolvendo bases de dados pequenas, ou desbalanceadas, com poucas amostras em determinada classe, é a especialização excessiva no conjunto de treinamento. Na problemática do presente trabalho, **D** pode facilmente passar a "decorar" quais são as imagens reais e passar a ignorar totalmente as imagens geradas por **G**, mesmo que **G** já esteja gerando imagens que sejam fidedignas à distribuição real. Para lidar com isso, a rede **D**, do método proposto, faz uso de uma camada de ruído gaussiano, apresentado na Seção 2.4. Essa camada, bloco azul na Figura 10, foi inserida no início da rede **D** e recebe a imagem de entrada, na qual adiciona um ruído gaussiano como média $\mu = 0$ e um baixo desvio padrão, $\sigma = 0,05$. Esse ruído adicionado irá aumentar a capacidade de generalização de **D**, evitando o *overfitting*. Dessa forma, aceitando as imagens, geradas por **G**, fiéis à distribuição real.
- **Minibatch-Discrimination:** Como mencionado na Seção 2.5, um problema comum em GANs é o colapso do modelo, mais especificamente da rede **G**. Quando se trata de bases de dados pequenas como um todo, ou poucas amostras em determinada classe, **G** pode passar a enganar **D** gerando sempre a mesma imagem. Como é proposto um método de *Oversampling*, é necessário que tal método seja capaz de gerar imagens parecidas com as reais, porém com alguma diversidade, para que assim possam contribuir na formação da região de decisão do classificador a ser utilizado. Para lidar com o problema do colapso, o método proposto incorpora na rede **D** a camada de *Minibatch-Discrimination*, representada pelo bloco verde, na Figura 10. A partir disso,

D percebe se **G** passa a gerar amostras sempre parecidas, e passa a negá-las, forçando **G** a gerar amostras mais diversas.

4.1.4 Etapas do Modelo

Nas próximas seções são apresentadas as etapas do treinamento do *Adversarial Oversampling*. Primeiramente, é feito um treinamento inicial. Logo após, inspirando-se nos métodos tradicionais de *Oversampling* citados no Capítulo 2, o foco passa a ser a geração de amostras consideradas difíceis. Para isso, há um segundo treinamento na intenção de fazer o gerador se especializar em gerar amostras que sejam consideradas difíceis e contribuam para o classificador. Esse segundo treinamento é feito utilizando a mesma rede adversária, já treinada. Por fim, há uma filtragem dos dados gerados para as classes minoritárias. Após a filtragem, os dados serão incluídos na base de dados original, balanceando as classes.

4.1.4.1 Treinamento Inicial

Na primeira etapa do AO, é feito o treinamento inicial recebendo o conjunto de dados desbalanceado **X**. A Rede Adversária recebe *batches* contendo a mesma proporção de todas as classes, com repetições dos exemplos das classes minoritárias. Pois nesse primeiro treinamento é necessário que ela seja capaz de aprender a distribuição dos dados e consiga traçar a superfície de decisão entre todas as classes do problema. Esse primeiro treinamento ocorre até que haja a convergência na do Discriminador (**D**) e do Gerador (**G**) da Rede Adversária. Essa convergência é dada quando o **D** e **G** erram e acertam na mesma proporção. O treinamento inicial é importante, pois a partir dele **G** será capaz de gerar amostras condicionadas às suas respectivas classes.

4.1.4.2 Segundo Treinamento - Especializando o Gerador

Após o treinamento inicial, a Rede Adversária, supostamente, já é capaz de gerar amostras sintéticas específicas para cada classe, ou seja, já aprendeu a gerar amostras condicionadas à distribuição das classes. O próximo passo é realizado por um segundo treinamento, que é executado para especializar o Gerador **G** na geração de amostras consideradas difíceis, ou seja, que estejam próximas às regiões de separação das classes. Essas amostras são ditas como difíceis, pois são visualmente parecidas com amostras de outras classes do problema. Por exemplo, considerando uma base de dados onde as classes são dígitos, e que a classe minoritária sobre a qual será realizado o *Oversampling* é a classe que representa o dígito 4. A partir do segundo treinamento, o modelo se especializa em gerar amostras do dígito

4, visualmente semelhantes a outra classe do problema. Observando a Figura 11, são exibidas amostras do dígito 4 similares a amostras do dígito 9. Essas amostras passariam a ter preferência na composição dos *batches* do segundo treinamento.

A motivação para esse procedimento vem dos métodos tradicionais, já apresentados na Seção 2.1.2, que concentram-se em gerar amostras difíceis durante o *Oversampling*. Tendo em vista que amostras sintéticas fáceis, por mais que estejam corretas e representem bem a classe em questão, não contribuem muito para o classificador. No segundo treinamento, a Rede Adversária se concentra nas amostras, pertencentes a classe minoritária, consideradas difíceis. Para isso, o segundo treinamento utiliza o bloco H , visto na Figura 12 e em parte da Figura 9, que é responsável por determinar quais amostras da classe minoritária serão apresentadas com mais frequência durante o treinamento, fazendo uso da função de entropia como medida de erro (MACÊDO et al., 2019).

$$H_m = - \sum_{c=1}^C \mathbf{D}(\mathbf{x}_m)_c \log \mathbf{D}(\mathbf{x}_m)_c \quad (4.5)$$

Como pode ser visto na Equação 4.5, a medida de entropia de uma amostra \mathbf{x}_m de uma classe minoritária será dada através da saída de do Discriminador \mathbf{D} , dado essa amostra. Como pode ser visto na Figura 10, \mathbf{D} possui duas saídas: Se a amostra é real, ou falsa, e qual as probabilidades dessa amostra pertencer a uma das c classes do problema. Logo, se for calculada a entropia de $\mathbf{D}(\mathbf{x}_m)$, utilizando a saída com as probabilidades das classes, será possível determinar o nível de confusão que a amostra \mathbf{x}_m causa em \mathbf{D} , com relação a qual classe ela pertence. Caso \mathbf{D} acerte a classe de \mathbf{x}_m , a medida de entropia é diretamente proporcional ao nível de dificuldade, pois mesmo acertando, para \mathbf{D} ainda há indícios de outras classes. Porém, se \mathbf{D} errar, a saída de entropia é inversamente proporcional a dificuldade, pois uma baixa entropia no erro significa que a rede esta errando a classe da amostra com mais certeza, e essa é uma amostra considerada difícil. Dessa forma, o *score* de dificuldade de uma amostra \mathbf{x}_m , H_{score_m} , será dado pela Equação 4.6, da seguinte forma:

$$H_{score_m} = \frac{H(\mathbf{D}(\mathbf{x}_m))^E}{\log C}, \quad (4.6)$$

onde E é 1, caso o discriminador acerte a classe da amostra, ou -1, caso contrário. H é a entropia calculada da saída da amostra \mathbf{x}_m no Discriminador \mathbf{D} , e C é a quantidade de classes do problema. Com isso, o H_{score_m} varia de 0 a 1, pois o valor máximo da entropia H é $\log C$. Logo, a chance da amostra \mathbf{x}_m compor o próximo *batch* de treinamento é um valor entre 0 e 1.

4.1.4.3 Filtragem das Amostras Sintéticas

Como já apresentado, as GANs estão no estado da arte para geração de amostras sintéticas. Porém, é importante que haja uma preocupação na seleção dessas amostras. Isso porque nem sempre elas irão manter a qualidade a medida que novas amostras são geradas por \mathbf{G} . Para isso, o método de filtragem proposto toma como base o trabalho trazido por (BHATTARAI et al., 2019), apresentado na Seção 2.6, para criar um filtro que possa tentar garantir a qualidade das amostras sintéticas que irão compor o *dataset* e balancear as classes. O filtro, representado pelo bloco F , Figura 13, consiste no uso do próprio \mathbf{D} para verificar a qualidade das amostras, \mathbf{x}_s , geradas por \mathbf{G} . Essa amostra é avaliada com relação à dois aspectos: *Class conditional probability (ccp)* e *Realism conditional probability (rcp)*.

$$Q_{score_{\mathbf{x}_s}} = ccp_{\mathbf{x}_s} \times rcp_{\mathbf{x}_s} \quad (4.7)$$

Onde,

$$ccp_{\mathbf{x}_s} = P(y_c | \mathbf{x}_s, \mathbf{D}_c) \quad (4.8)$$

$$rcp_{\mathbf{x}_s} = \mathbf{D}(\mathbf{x}_s) \quad (4.9)$$

Como pode-se ver na Equação 4.7, o *score* de qualidade, $Q_{score_{\mathbf{x}_s}}$, da amostra sintética \mathbf{x}_s , é dado pelo produto dos *scores* de realismo e de classe, $rcp_{\mathbf{x}_s}$ e $ccp_{\mathbf{x}_s}$, respectivamente. Com $rcp_{\mathbf{x}_s}$, será possível saber o quão real é a amostra sintética gerada \mathbf{x}_s , com base no discriminador \mathbf{D} treinado. Também com base em \mathbf{D} , será possível saber o quão a amostra \mathbf{x}_s é similar às amostras pertencentes a classe c , através de $ccp_{\mathbf{x}_s}$, no qual é utilizada a saída da rede auxiliar do Discriminador \mathbf{D} , que possui a informação da distribuição das classes do problema. A ideia por trás do produto entre $rcp_{\mathbf{x}_s}$ e $ccp_{\mathbf{x}_s}$, é que, para uma amostra ser considerada útil, ela precisa seguir a distribuição das amostras originais da base, e também seguir distribuição real da classe em questão. Dessa forma, se $rcp_{\mathbf{x}_s}$, ou $ccp_{\mathbf{x}_s}$, for 0, o $Q_{score_{\mathbf{x}_s}}$ passará a ser 0 e \mathbf{x}_s não terá chances de compor o conjunto de treinamento. Caso contrário, quanto maior o $Q_{score_{\mathbf{x}_s}}$, maior a chance de \mathbf{x}_s compor a base de dados.

O fluxo de uso do filtro no AO pode ser visto na arquitetura geral do método, exibido na Figura 9. Como pode-se ver, o filtro F é a ultima etapa do método propósito, então a partir dele as amostras da classe minoritária irão ser inseridas na base de dados para balancear as classes.

Figura 11 – Plot 2D de amostras, visualmente similares, dos dígitos 4 e 9.

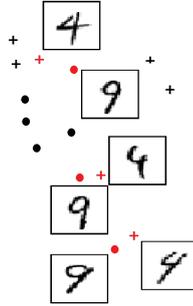


Figura 12 – Bloco H que representa a etapa de Entropia para determinar a dificuldade das amostras.

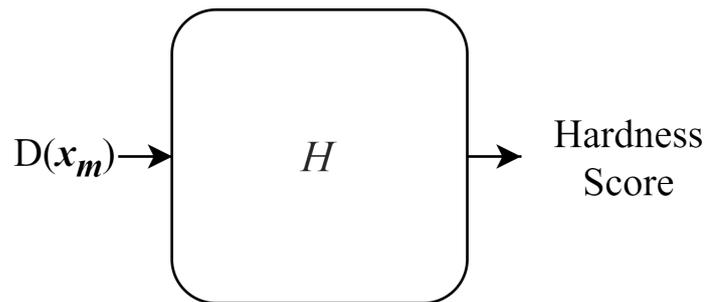
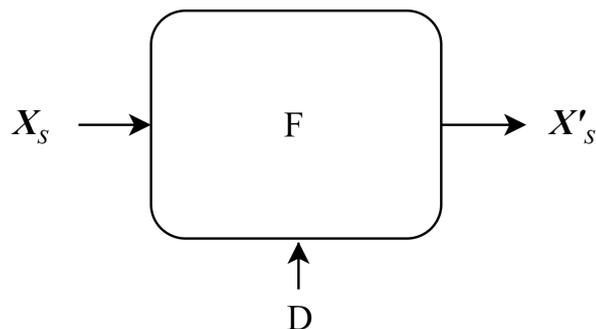


Figura 13 – Bloco que representa a etapa de filtragem das amostras sintéticas geradas pelas Redes Adversárias. Recebendo um *batch* de amostras sintéticas \mathbf{X}_s e o Discriminador \mathbf{D} , já treinado. Como saída, \mathbf{X}'_s são as imagens que passaram pela filtragem. Consistindo na etapa final do *Adversarial Oversampling*.



5 EXPERIMENTOS

Os experimentos consistem primeiramente em avaliar a qualidade das imagens geradas pelo método proposto em duas perspectivas: Qualidade visual das amostras geradas e contribuição na taxa de acerto do classificador. O método proposto foi comparando com métodos tradicionais e amplamente utilizados para a tarefa de *Oversampling*: SMOTE, ADASYN, KSMOTE, BSMOTE e SVMSMOTE. Além desses métodos, a proposta também foi comparada com o uso da AC-GAN original.

5.1 BASES DE DADOS

Para avaliação dos métodos de *Oversampling*, foram utilizados bases de dados envolvendo imagens, com tamanhos e características diferentes. A maioria das bases contém imagens médicas, dentre os quais estão: *Breast* (CRUZ-ROA et al., 2014), *Heartbeat* (Moody; Mark, 2001), *Glaucoma* (ZHANG et al., 2010), *XRay* (KERMANY; ZHANG; GOLDBAUM, 2018) e *Skin Cancer*¹. A seguir, essas bases de dados serão brevemente descritas. Uma sumarização das informações dessas bases pode ser vista na Tabela 2.

5.1.1 *Breast*

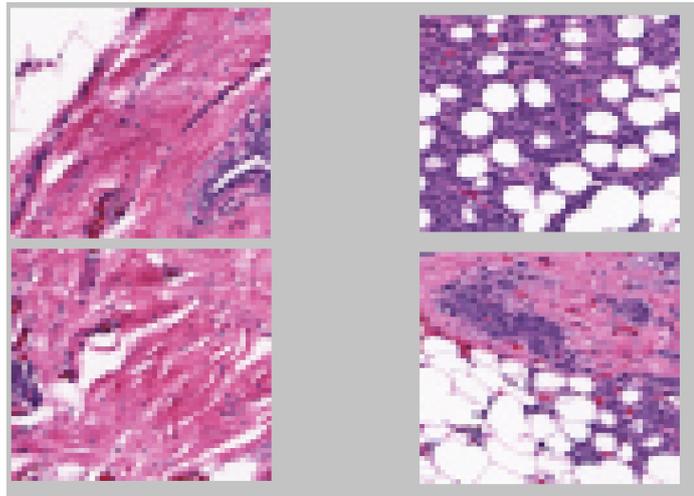
A primeira utilizada envolve o problema de Câncer de mama, mais especificamente o tipo Carcinoma Ductal Invasivo (CDI). A base de dados (CRUZ-ROA et al., 2014)) originalmente é formada por 162 imagens de tipos de Câncer, digitalizadas com zoom de 40x. A partir dessas imagens, foram extraídos 277.524 *patches* de tamanho 50×50 , resultando em 198.738 da classe majoritária, contra 78.787 da classe minoritária. No presente trabalho, será utilizado a versão da base de dados que contém os *patches*. Na Figura 14 pode ser visto à esquerda dois exemplos da classe negativa (Sem CDI) e à direita dois exemplos da classe positiva (Com CDI).

5.1.2 *XRay*

Uma outra base de dados utilizada é composta por imagens de Raio-X do tórax de pacientes do Centro Médico de Guangzhou (KERMANY; ZHANG; GOLDBAUM, 2018). Essas imagens são categorizadas em duas classes: Normal e Pneumonia. A base de dados contém 5.863 imagens de Raio-X, das quais 1341 são da classe Normal, e

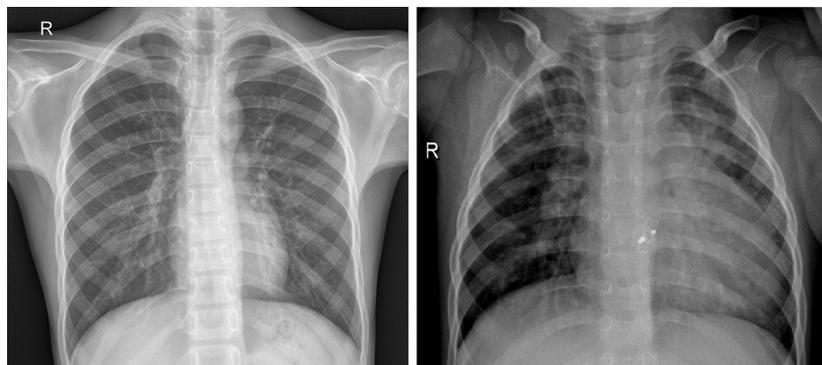
¹ <https://www.isic-archive.com/>

Figura 14 – Exemplo de imagens da base de dados *Breast*.



3875 da classe Pneumonia. As imagens são representadas em tons e cinza e possuem as dimensões padronizadas em 112×112 . Dois exemplos da base podem ser vistos na Figura 15, na esquerda contém o exemplo de um Raio-X categorizado como Normal, e na direita um exemplo de um Raio-X categorizado como Pneumonia.

Figura 15 – Exemplo de imagens da base de dados *XRay*.

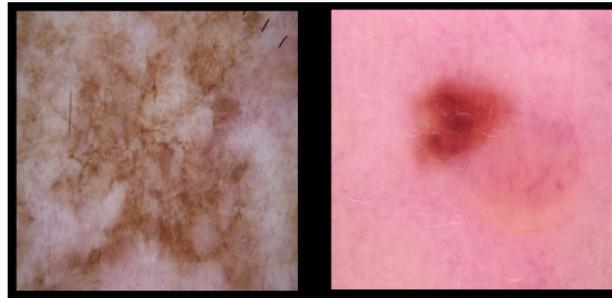


5.1.3 Skin Cancer

A próxima base de dados utilizada envolve a problemática de Câncer de Pele. A base em questão foi disponibilizada pelo ISIC (*International Skin Imaging Collaboration*)² e consiste em 2000 imagens. A divisão entre as classes é: 1626 imagens da classe Normal (benigno) e 374 imagens da classe Anormal (maligno). As imagens são representadas em *rgb* e possuem as dimensões $3 \times 224 \times 224$. Dois exemplos dessa base podem ser vistos na 16, na qual a direita contém um exemplo positivo (Glaucoma), e a esquerda um exemplo negativo (Normal).

² <https://www.isic-archive.com/>

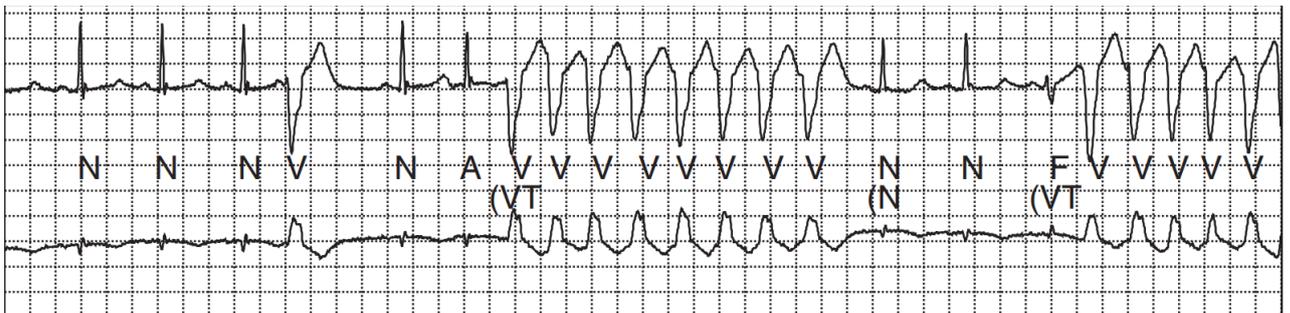
Figura 16 – Exemplo de imagens da base de dados *Skin Cancer*.



5.1.4 *Heartbeat - MITBIH*

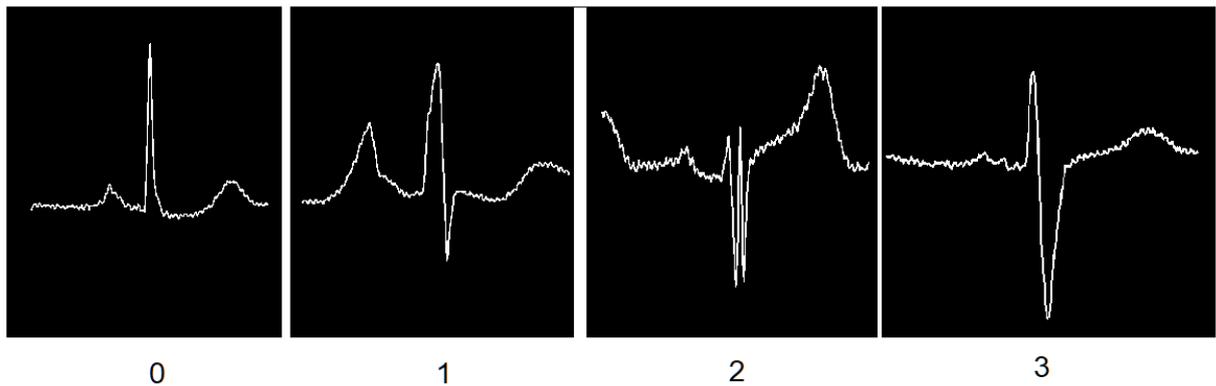
Uma base de dados de batimentos cardíacos também foi utilizada, o MIT-BIH Arrhythmia Database (Moody; Mark, 2001). Essa base consiste em 48 trechos de meia hora de gravações ambulatoriais de Eletrocardiograma (ECG) obtidas de 47 indivíduos estudados pelo Laboratório BIH de Arritmia, entre os anos 1975 e 1979. O sinal foi coletado por dois canais e com a frequência determinada em 360hz. A base em original é representada por séries temporais 1D, possuindo marcações em cada batimento, as quais representam os tipos. Um exemplo pode ser visto na Figura 17, a qual possui as marcações N (*Normal*), A (*Atrial premature beat*), F (*Ventricular fusion beat*) e V (*Ventricular premature beat*).

Figura 17 – Exemplo do sinal de ECG da base de dados *Heartbeat*.



Por se tratar de uma base envolvendo séries temporais (1D), e como o presente trabalho lida com o desbalanceamento em bases de dados composta por imagens, foi realizado um pré-processamento que passou a base para ser representada por imagens. Cada imagem consistiu em *plots* 2D de cada batimento, com sua respectiva classe. Após a aquisição do sinal ECG 1D, um pré-processamento foi feito para coletar um intervalo do sinal, que represente um batimento cardíaco. Após isso, esse intervalo foi plotado em uma imagem binária 2D, com dimensões 112×112 . A Figura 18 mostra a base após o processamento, com exemplos de cada uma das suas quatro classes.

Figura 18 – Exemplo de imagens da base de dados *Heartbeat*, das quatro classes existentes, após pré-processamento.



5.1.5 *Glaucome - ORIGA*

A próxima base utilizada contém imagens do fundo da retina, as quais caracterizam o problema de glaucoma. A base *Glaucome (ORIGA)* (ZHANG et al., 2010) é composta imagens coloridas com dimensão $3 \times 3072 \times 2048$. Nessa base, 482 imagens são pertencentes a classe negativa (Normal) e somente 168 imagens são pertencentes a classe positiva (Glaucoma). Dois exemplos dessa base podem ser vistos na Figura 19, na qual à direita contém um exemplo positivo (Glaucoma), e à esquerda um exemplo negativo (Normal).

Figura 19 – Exemplo de imagens da base de dados *Glaucome*.

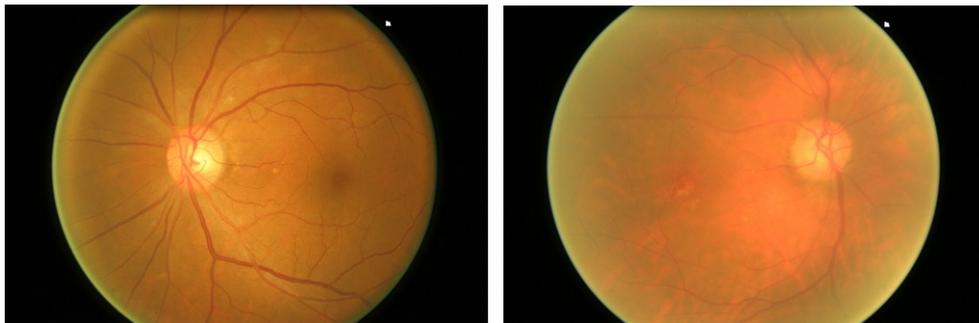


Tabela 2 – Sumarização das bases de dados utilizadas

Base de Dados	Dimensão	Exemplos por Classe
Breast	(3, 32, 32)	(0 : 18591 1 : 9214)
Heartbeat	(1, 128, 128)	(0 : 1000 1 : 1000 2 : 200 3 : 200)
Glaucome	(3, 224, 224)	(0 : 482 1 : 168)
XRy	(1, 112, 112)	(0 : 134 1 : 3875)
Skin Cancer	(3, 224, 224)	(0 : 374 1 : 1626)

5.2 MEDIDAS DE AVALIAÇÃO

Para avaliar modelos sobre conjuntos de testes desbalanceados, a escolha da medida de acerto é fundamental. Como a base de teste possui poucas amostras da classe minoritária, um modelo que ignore essa classe pode ser favorecido caso se utilize uma medida de avaliação tradicional, como a acurácia. Isso acontece pois, considerando que a proporção da classe minoritária seja somente 20%, um modelo que dê como resposta somente a classe majoritária terá 80% de acerto, o que é aceitável em muitos casos. Porém, todas as amostras que pertencem da classe minoritária iriam ser erradas, o que faz essa medida utilizada ser inútil.

Tendo em vista o problema acima, para avaliação do método proposto, bem como comparação do mesmo com os tradicionais, foram escolhidas duas medidas amplamente difundidas no estado da arte: Média Geométrica e a área sob a curva ROC (AUC-ROC). A seguir, essas medidas escolhidas serão descritas brevemente.

5.2.1 Média Geométrica

Média Geométrica (G_{mean}) é uma métrica muito útil para avaliar modelos em bases de dados desbalanceadas (BARANDELA et al., 2003). É calculada através da raiz do produto entre o *Recall* e o *Precision*, ou seja, se ao menos uma classe não for reconhecida pelo modelo, o resultado é 0. Essa fator é importante para penalizar um classificador que retorne sempre uma única classe do problema.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (5.1)$$

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5.2)$$

$$G_{mean} = \sqrt{precision \times recall} \quad (5.3)$$

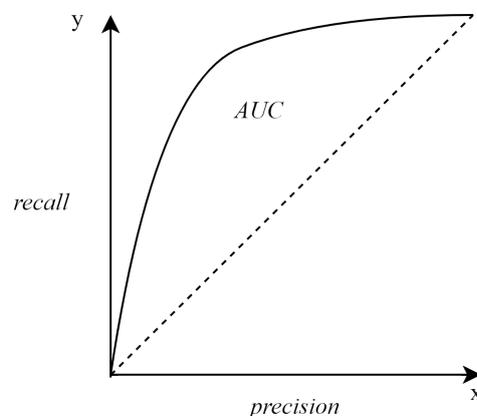
5.2.2 AUC-ROC

A área sob a curva ROC (BRADLEY, 1997) é uma medida de performance que considera o ranking do classificador, ou seja, é calculada a partir das probabilidades, ao contrário da média geométrica, mencionada anteriormente, que considera somente o valor absoluto, se a amostra pertence ou não a uma determinada classe (0 ou 1). *ROC* é a curva de probabilidade e *AUC* a o grau de separabilidade das classes positivas e negativas. Maior *AUC* médio significa que o modelo em questão é melhor na capacidade de distinguir entre as classes do problema. Para lidar com um possível desequilíbrio na proporção de classes, a curva é construída utilizando *precision* e *recall*, nos eixos *x* e *y* no gráfico da Figura 20.

$$\text{recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (5.4)$$

$$\text{precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (5.5)$$

Figura 20 – Exemplo da área sobre a curva. No eixo *x* há o *precision* e no eixo *y* o *recall*.



5.3 DIVISÃO DOS CONJUNTOS DE DADOS

Uma parte de extrema importância da avaliação de modelos de aprendizagem de máquina é a divisão da base de dados em amostras que irão compor o conjunto de treinamento e amostras que irão compor o conjunto de testes. No presente trabalho, os métodos de *Oversampling* utilizaram somente o conjunto de treino para se basear nas suas respectivas gerações de amostras sintéticas. Dessa forma, o conjunto de testes foi utilizado somente pelo classificador final, para avaliá-lo e indiretamente avaliar os métodos de *Oversampling*. A base de treinamento e teste foi

separada utilizando amostragens aleatórias, estratificadas e repetição, mantendo a proporção de 90% para treinamento e 10% para teste.

5.4 CLASSIFICADORES

Como o presente trabalho lida com a bases de dados envolvendo imagens, os classificadores utilizados são Redes Neurais Convolucionais (*Convolutional Neural Networks* (CNN)). As arquiteturas das CNNs escolhidas estão amplamente difundidas no estado da arte e possuem disponíveis materiais para estudo e implementações, testadas e validadas. São elas:

- DenseNet (HUANG; LIU; WEINBERGER, 2016);
- ResNet50 (HE et al., 2015);
- VGG16 (SIMONYAN; ZISSERMAN, 2014).

5.5 AVALIAÇÕES

A seguir, estão descritos os experimentos que avaliaram o AO, como método de *Oversampling*, e também o comparou com os demais métodos trazidos no presente trabalho. Os objetivos de cada experimento foram apresentados, bem como as respectivas metodologias experimentais adotadas.

5.5.1 Contribuição do AO para o Acerto da Classe Minoritária

O primeiro experimento realizado tem o objetivo de avaliar o quão as amostras sintéticas geradas pelo AO contribuem para o aumento da taxa de acerto envolvendo a classe minoritária. Para isso, foram inseridas, de forma incremental, amostras sintéticas da classe minoritária, e foi observado como se comportou a taxa de acerto na classe. As amostras da classe minoritária foram inseridas na proporção até que o balanceamento da base de dados fosse alcançado. A métrica utilizada é o acerto na classe minoritária dado pela Equação 5.6.

$$ACC_m = \frac{TP_m}{TP_m + FP_m} \quad (5.6)$$

Onde ACC_m é a taxa de acerto da classe minoritária, TP_m é a quantidade de vezes em que a classe minoritária foi classificada corretamente e FP_m é a quantidade de vezes em que a classe minoritária foi classificada de forma incorreta.

5.5.2 Impacto da Filtragem no AO

O próximo experimento realizado teve o objetivo de verificar o impacto da etapa de filtragem proposta, apresentada na Seção 4.1.4.3, para geração de amostras sintéticas por parte do AO. Para isso, o experimento descrito na Seção 5.5.1 foi executado novamente para o AO sem a etapa de filtragem. Dessa forma, foram comparados os resultados do AO utilizando a filtragem proposta, com o AO sem essa etapa de filtragem. A medida de desempenho utilizada foi o acerto na classe minoritária, dado pela Equação 5.6.

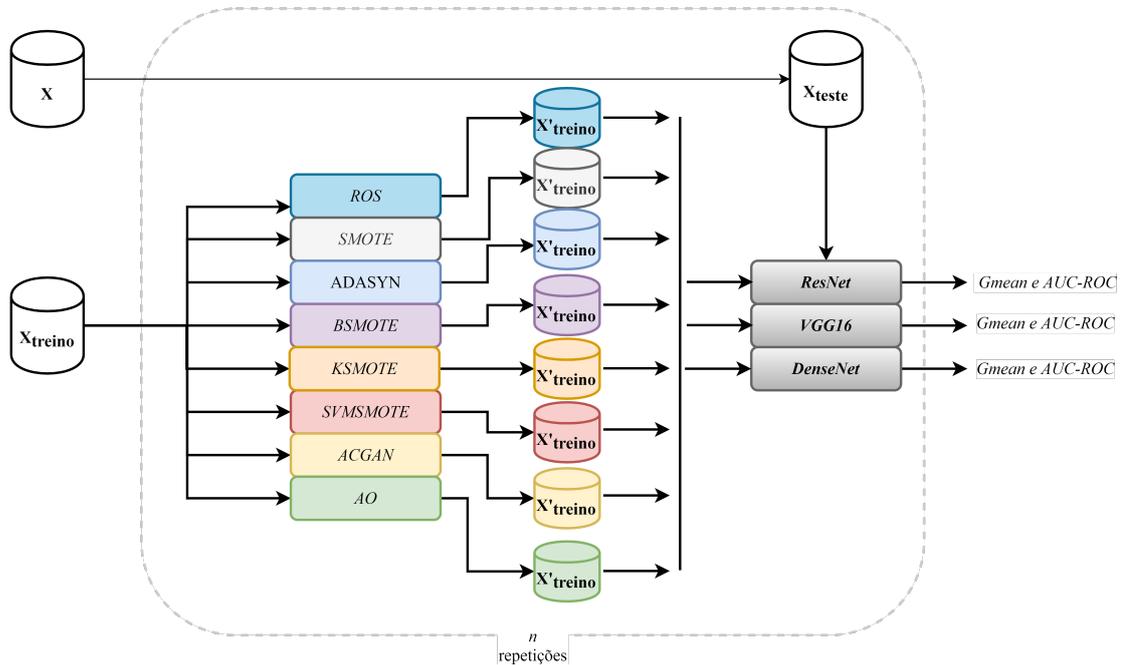
5.5.3 Aspecto Visual das Amostras

Uma outra análise consistiu em observar o aspecto visual das imagens geradas pelos métodos de *Oversampling*. Esse experimento foi realizado comparando alguns exemplos de amostras geradas pelo método proposto e pelo SMOTE, o qual utiliza uma interpolação e não realiza essa geração de acordo com o aprendizado dos dados. As geradas foram plotadas e foram observados os aspectos visuais. Essa análise foi feita para, de forma intuitiva, observar as diferenças entre as imagens geradas através das convoluções, com o aprendizado das GANs, das imagens geradas sem um aprendizado prévio, somente pela interpolação feita no SMOTE, e qual das formas preserva mais características importantes da imagem.

5.5.4 Comparativo entre os Métodos Tradicionais de *Oversampling*

O último experimento realizado consistiu na comparação o *Adversarial Oversampling* com as técnicas de *Oversampling* tradicionalmente utilizadas: ROS, SMOTE, ADASYN, KSMOTE, BSMOTE, SVMSMOTE. Além disso, o AO também foi comparado com as AC-GAN. Na Figura 21 pode-se ver a metodologia experimental utilizada. Primeiramente, a base de dados desbalanceada, \mathbf{X}_{treino} , foi utilizada para o treinamento dos métodos de *Oversampling* abordados. Após isso, cada método gerou seu respectivo \mathbf{X}'_{treino} , que consiste na base de dados balanceada, com as amostras geradas pelo respectivo método. A etapa de avaliação de um método de *Oversampling* consistiu em treinar um classificador utilizando a base de dados balanceada por tal método, e aferir o desempenho do classificador utilizando o conjunto de teste, \mathbf{X}_{teste} , até então não visto. Cada método de *Oversampling* foi testado com três classificadores diferentes: *ResNet*, *DenseNet* e *VGG16*. Esse procedimento foi repetido n (10) vezes e os conjuntos de treinamento foram determinados através de n amostragens aleatórias, sem repetição e mantendo a proporção original das classes. Por fim, foram utilizadas as medidas de desempenho AUC-ROC e G_{mean} , considerando suas respectivas médias das n repetições.

Figura 21 – Fluxo experimental da comparação entre o método proposto e os demais métodos de *Oversampling*



5.6 AMBIENTE

Quando se trata de bases de dados envolvendo imagens e classificadores como as CNN, que são consideradas arquiteturas profundas, é inevitável o alto custo computacional. Dessa forma, é necessário que haja uma infraestrutura adequada para tornar os experimentos viáveis. Para realização dos experimentos, foi utilizada uma infraestrutura (Tabela 3), contando com uma *gpu* que tornou possível paralelizar algumas operações através da biblioteca CUDA³. A infraestrutura tornou viável a execução, mesmo que demorada, dos experimentos em todas as bases de dados citadas.

Tabela 3 – Infra-estrutura utilizada para realização dos experimentos.

Framework	Tensorflow 1.13.1 e Keras 2.2.4
CUDA	CUDA 10.0.130
Linguagem	Python 3.6
GPU	NVIDIA GeForce GTX 1080
Núcleos	8
Memória	62GB
Armazenamento	60GB

³ <<https://developer.nvidia.com/cuda-zone>>

5.7 CONFIGURAÇÃO PARAMÉTRICA PARA COMPARAÇÃO DOS MÉTODOS

Métodos de *Oversampling*: Para tornar a aplicação dos métodos mais prática, foi adotada a biblioteca *imblearn* (LEMAÎTRE; NOGUEIRA; ARIDAS, 2017), que contém implementação de vários métodos para lidar com o problema de desbalanceamento. Dentre esses métodos, estão os tradicionais de *Oversampling*, aqui utilizados: ROS, SMOTE, ADASYN, KSMOTE, BSMOTE, SVMSMOTE. Todas as configurações paramétricas foram mantidas como as já implementadas na biblioteca. Mais detalhes quanto às configurações paramétricas, podem ser vistas na documentação oficial do *imblearn*, na seção de *Oversampling* ⁴.

Com relação aos métodos baseados nas AC-GAN, a configuração paramétrica consistiu na seguinte: Como otimizador foi utilizado o Adam (KINGMA; BA, 2014), com taxa de aprendizagem ajustada para 0.002; O tamanho do *batch* de treinamento variou de acordo com o tamanho do conjunto de treinamento \mathbf{X} , e também de acordo com a quantidade de amostras da classe minoritária. A quantidade máxima de épocas de treinamento definidas foi de 30 mil. Os treinamentos foram interrompidos até que \mathbf{D} e \mathbf{G} errem e acertem na mesma proporção, já que trata-se de um aprendizado adversário em um jogo de soma zero. Esses parâmetros foram definidos de forma empírica, através de experimentos prévios.

Classificadores: Para os classificadores, a quantidade máxima de épocas de treinamento foi de 500 e a *loss* no conjunto de validação foi monitorada para determinar quando o treinamento deveria ser interrompido. Isso empiricamente determina que a rede não está avançando na aprendizagem e o treinamento pode ser interrompido. A escolha do tamanho do *batch* também variou de acordo com a dimensão das imagens no conjunto de treinamento \mathbf{X} . Essa configuração foi baseada em experimentos anteriores, e foi mantida para todos os métodos de *Oversampling* avaliados.

⁴ <https://imbalanced-learn.readthedocs.io/en/stable/api.html#module-imblearn.over_sampling>

6 RESULTADOS E DISCUSSÕES

Nas próximas seções são apresentados os resultados obtidos através dos experimentos mencionados no Capítulo 5. Primeiramente, os resultados trazidos na Seção 6.1 expressam a comparação da qualidade visual das amostras geradas pelo *Adversarial Oversampling* (AO) com as geradas através do SMOTE, o qual realiza uma interpolação. Posteriormente, na Seção 6.2 é exibido o quanto as amostras sintéticas geradas pelo AO contribuíram para o aumento do acerto na classe minoritária. Por fim, a Seção 6.4 traz os resultados, em diversas bases de imagens, da comparação do método proposto AO com os demais métodos para *Oversampling* aqui apresentados.

6.1 QUALIDADE VISUAL

Os métodos de *Oversampling* foram avaliados quanto ao aspecto visual das amostras geradas. Na Figura 22, temos amostras geradas utilizando uma base envolvendo dígitos, no caso, referentes aos dígitos 0 e 4. Na esquerda temos a geração feita pelo SMOTE e, na direita, temos a geração pelo método proposto AO. Como pode ser visto, a geração através do método AO é mais fiel à representação dos dígitos 0 e 4. Esse fator pode ter ocorrido porque a geração dos métodos tradicionais ocorre através de interpolações entre imagens existentes, sem um processo de aprendizado, diferentemente do AO.

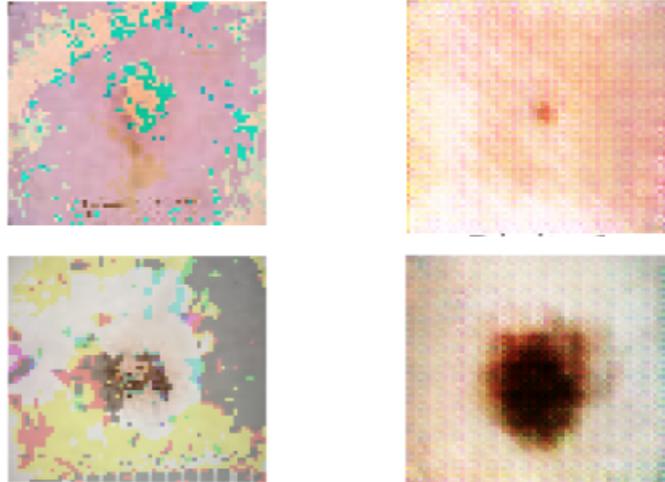
Figura 22 – Geração de amostras sintéticas dos dígitos 0 e 4 utilizando os métodos tradicionais (esquerda) e utilizando o AO (direita).



Um outro exemplo da qualidade visual das imagens geradas pode ser visto na Figura 23, onde são exibidas as amostras geradas utilizando a base *Skin Cancer*. Na esquerda há uma geração feita pelo SMOTE, e na direita pelo método AO. Como

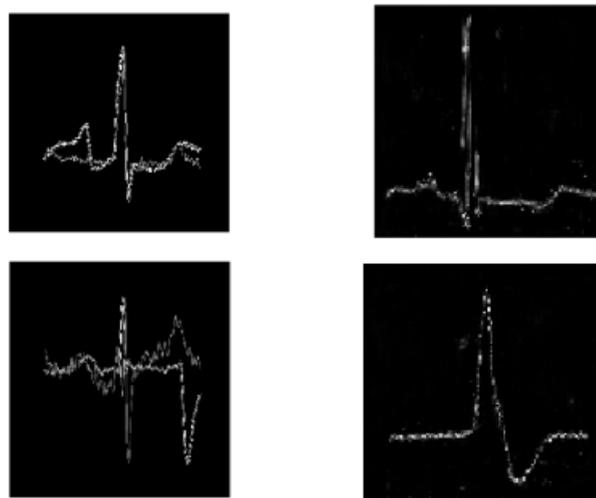
pode ser notado, a geração pelo AO pareceu ser mais fiel à representação dos sinais de tumores. Sendo possível observar aspectos de cor e bordas, importantes para classificar as imagens de tumores em benignos (superior) ou malignos (inferior).

Figura 23 – Geração de amostras sintéticas de imagens da base *Skin Cancer* utilizando os métodos tradicionais (esquerda) e utilizando o AO (direita).



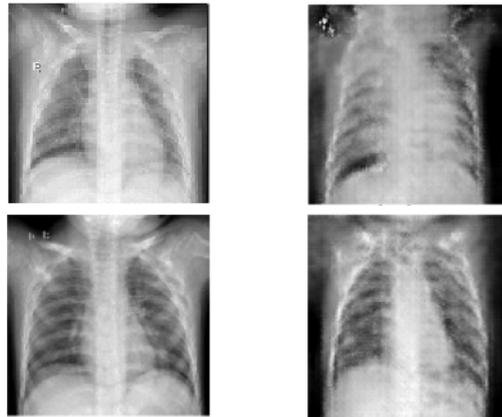
Na Figura 24, é exibido o resultado das imagens geradas utilizando a base *Heartbeat*, referentes aos sinal que representam o batimento cardíaco. Do lado esquerdo da Figura 24 tem-se a geração feita pelo método SMOTE, e na direita a geração feita gerado pelo método proposto AO. Observando as imagens geradas pelos métodos baseados no SMOTE, pode-se notar que há uma sobreposições dos sinais de ECG. Já observando a geração pelo AO, nota-se uma maior fidelidade à representação de um sinais de ECG, onde, como é de se esperar, é observado somente um sinal.

Figura 24 – Geração de amostras sintéticas de imagens da base *Heartbeat* utilizando os métodos tradicionais (esquerda) e utilizando o AO (direita).



Por fim, foi observado a qualidade das imagens sintéticas geradas da base *XRay*. Pode-se ver que, como as imagens são bem centralizadas, os métodos baseados no SMOTE obtiveram um bom resultado visual. O mesmo ocorreu com o AO, que com poucas épocas já conseguiu gerar fiéis às originais.

Figura 25 – Geração de amostras sintéticas de imagens da base *XRay* utilizando os métodos tradicionais (esquerda) e utilizando o AO (direita).



6.2 AMOSTRAS GERADAS PELO AO E PRECISÃO

Nessa Seção pode ser observada a contribuição do AO para a precisão na classe minoritária. É visto o quão as amostras sintéticas, da classe minoritária, geradas pelo AO contribui para o classificador. Os resultados exibidos nessa Seção são relacionados ao experimento descrito na Seção 5.5.1 e são exibidos em imagens dos gráficos da variação da precisão à medida que são inseridas amostras sintéticas. Os resultados a serem apresentados são referentes a uma execução a *batch* de amostras geradas inseridas no treinamento.

Primeiramente, pode-se ver na Figura 26 que o método proposto conseguiu contribuir gerando imagens sintéticas de radiografias para a classe minoritária. Houve um aumento em torno de 5% na precisão da classe. Porém, é perceptível que nem sempre a maior quantidade de amostras inseridas significa em um maior desempenho.

Lidando com a base de dados *Skin Cancer*, pode-se ver no gráfico da precisão na classe minoritária, Figura 27, um aumento de quase 10% ao inserir amostras sintéticas geradas pelo AO.

Considerando a base de batimentos cardíacos, *Heartbeat*, pode-se observar um aumento bem consistente da precisão nas classes minoritárias 2 e 3, nos gráficos da Figuras 28 e 29, respectivamente.

Figura 26 – Variação da precisão da classe minoritária na base *XRay*, à medida que são inseridas amostras sintéticas geradas pelo AO.

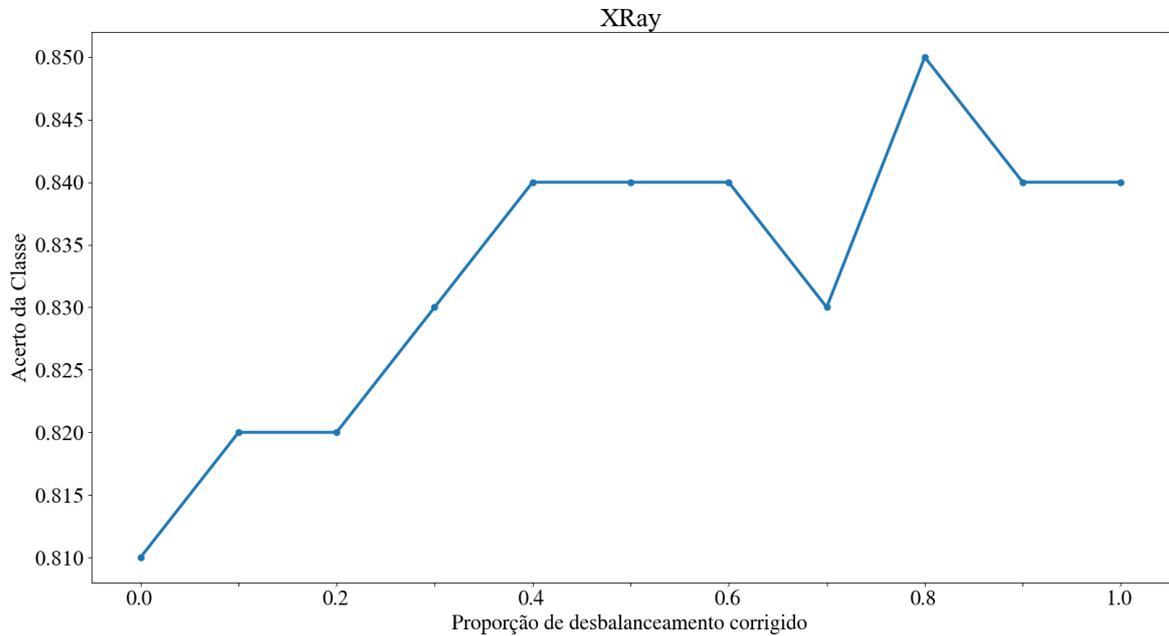
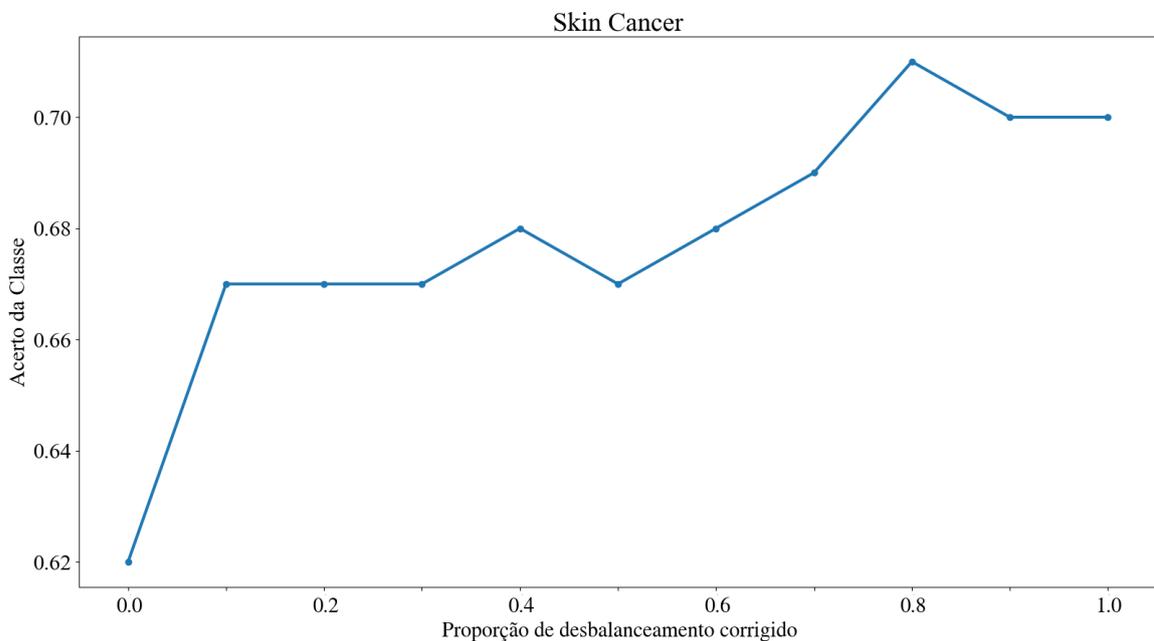


Figura 27 – Variação da precisão da classe minoritária na base *Skin Cancer*, à medida que são inseridas amostras sintéticas geradas pelo AO.



Na Figura 30, pode-se observar, na base de dados *Glaucome*, que o aumento na precisão da classe minoritária não teve um crescimento imediato a partir das primeiras amostras sintéticas inseridas. Porém, o crescimento começou logo após as primeiras 20 amostras e, empiricamente, mostrou uma contribuição dessas amostras na taxa de acerto da classe minoritária.

Figura 28 – Variação da precisão da classe minoritária 2 na base *Heartbeat*, à medida que são inseridas amostras sintéticas geradas pelo AO.

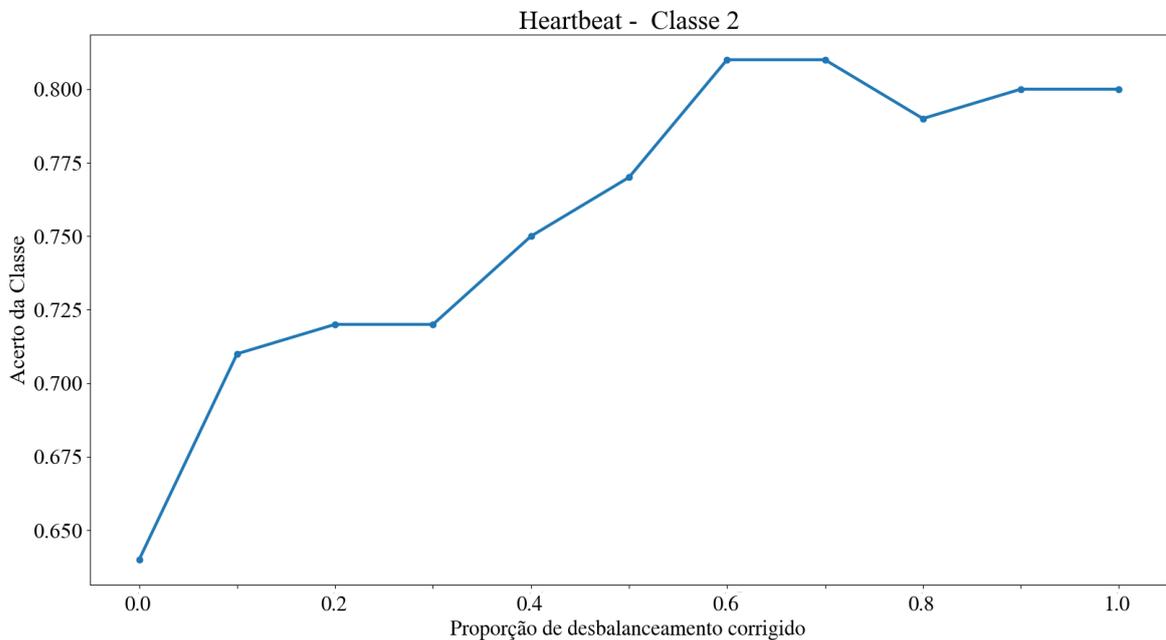
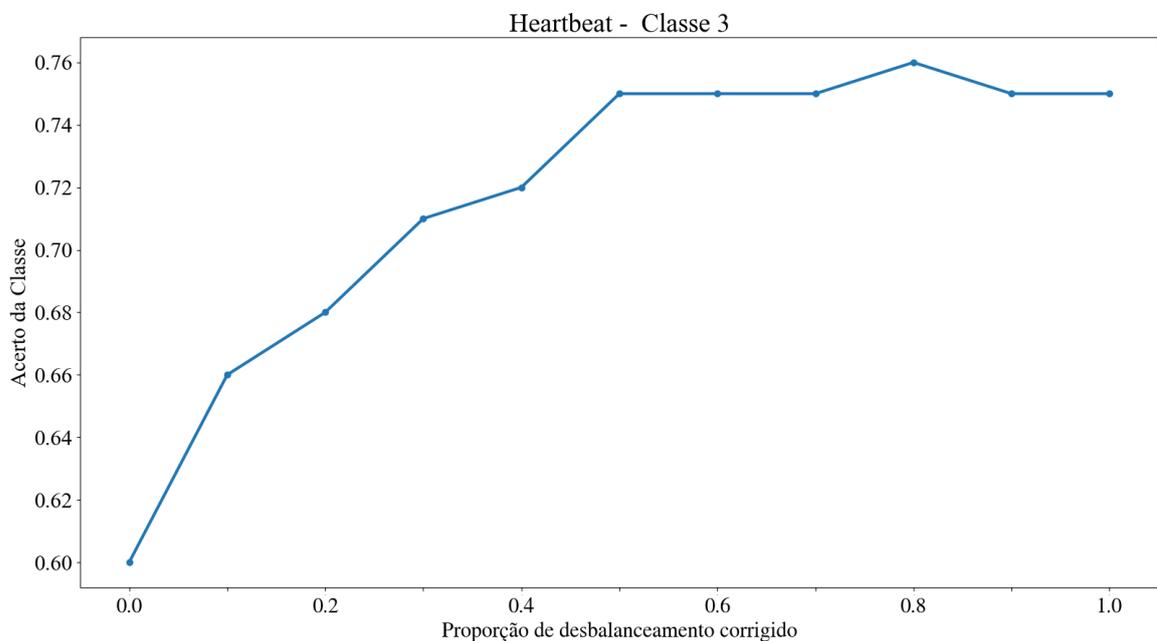


Figura 29 – Variação da precisão da classe minoritária 3 na base *Heartbeat*, à medida que são inseridas amostras sintéticas geradas pelo AO.



A precisão da classe minoritária na base de dados *Breast*, expressa no gráfico da Figura 31, um crescimento inconsistente até que uma boa quantidade de amostras sintéticas fosse inserida e, logo após isso, houve um crescimento considerável de quase 4%.

De forma geral, foi possível notar, através dos experimentos e dos resultados

Figura 30 – Variação da precisão da classe minoritária na base *Glaucome*, à medida que são inseridas amostras sintéticas geradas pelo AO.

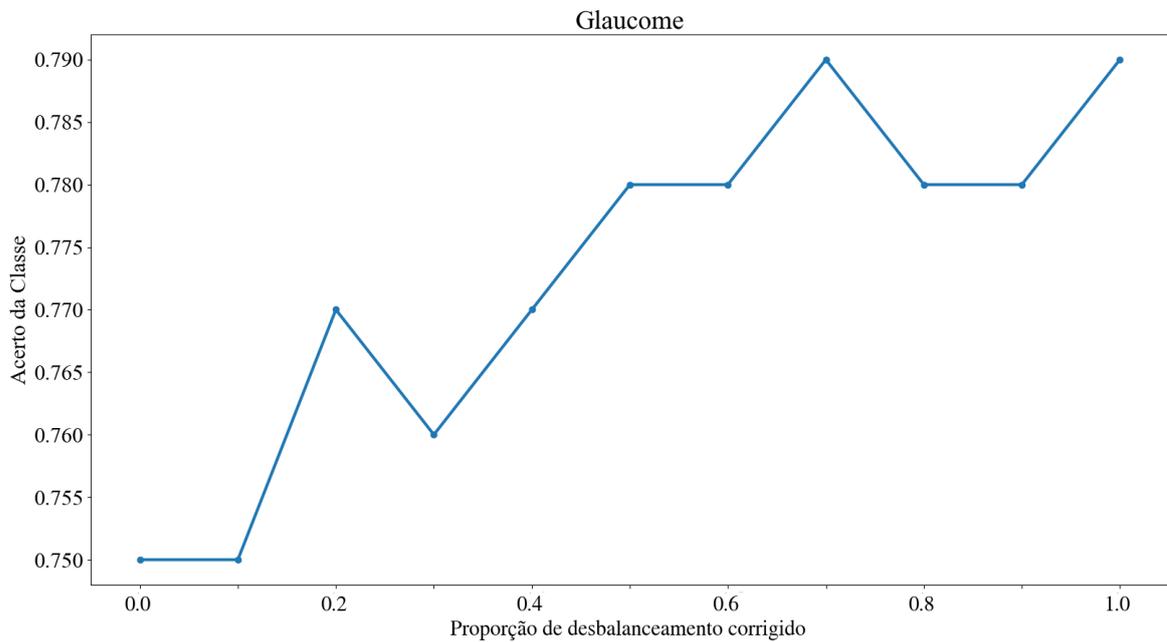
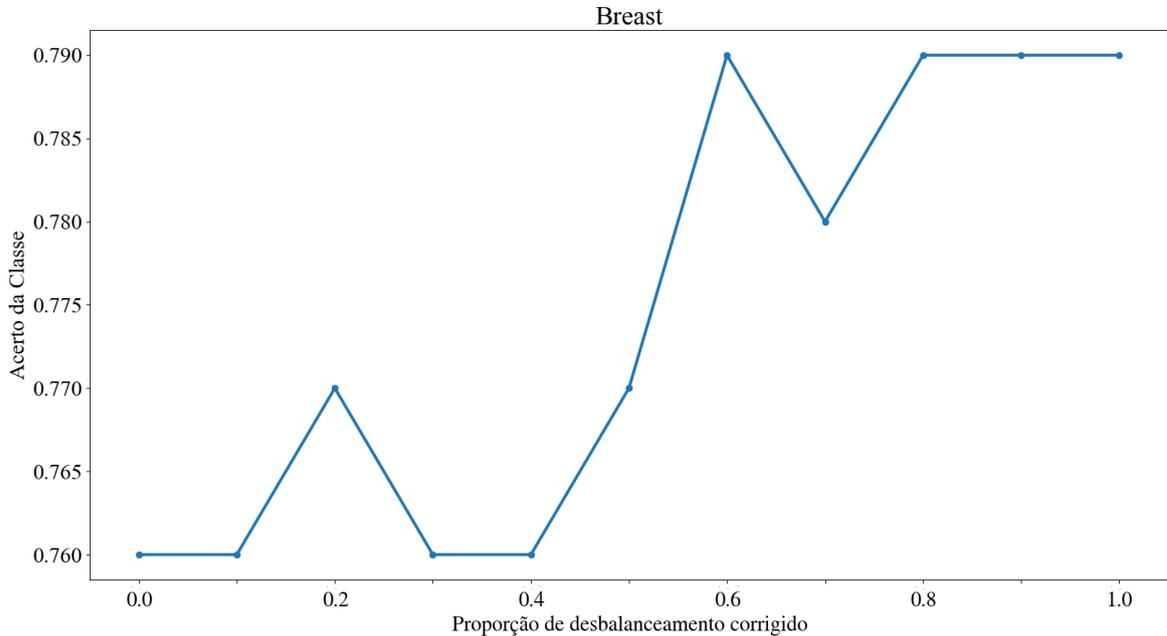


Figura 31 – Variação da precisão da classe minoritária na base *Breast*, à medida que são inseridas amostras sintéticas geradas pelo AO.



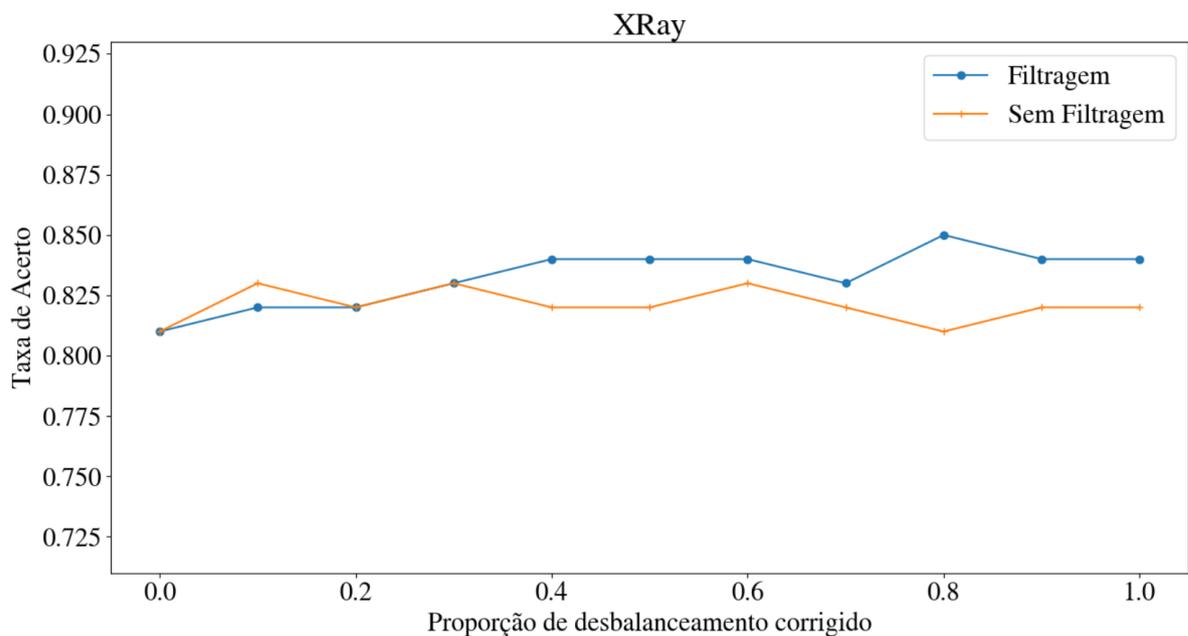
obtidos, que o método proposto colaborou com uma melhora na precisão da classe minoritária, ao gerar amostras para a mesma durante o treinamento. Um outro fator que pode-se notar é que nem sempre uma maior quantidade de amostras sintéticas inseridas significa uma maior precisão.

6.3 IMPACTO DA FILTRAGEM NO AO

Os resultados apresentados a seguir estão relacionados ao experimento descrito na Seção 5.5.2. São exibidos em forma de gráficos contendo a variação da precisão na classe minoritária, à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem uso da etapa de filtragem. Os resultados a serem apresentados são referentes a uma execução a *batch* de amostras geradas inseridas no treinamento, com e sem a etapa de filtragem.

Primeiramente, pode-se observar, no gráfico da Figura 32, os resultados na base de dados *XRay*. É possível notar que o uso da filtragem conseguiu manter a geração de amostras estável, no que diz respeito à precisão, nas primeiras amostras inseridas. Porém, essa melhoria só foi mantida na versão que fez uso da filtragem proposta.

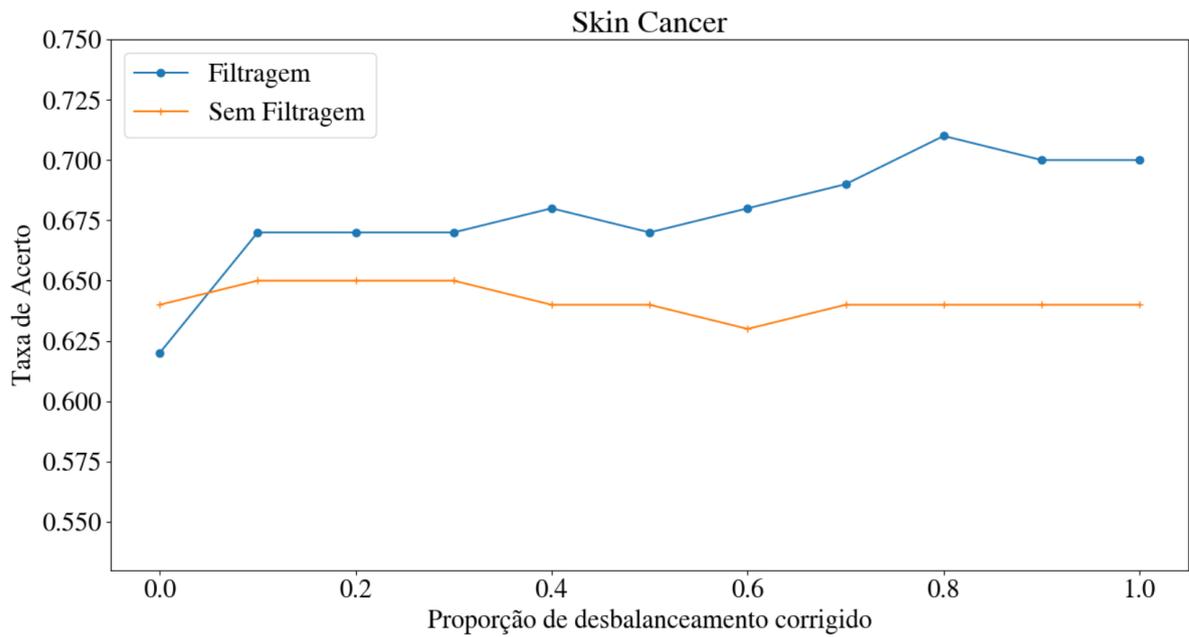
Figura 32 – Variação da precisão da classe minoritária na base *XRay*, à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.



No gráfico da Figura 33, pode-se observar os resultados na base *Skin Cancer*. De uma forma mais clara, é possível observar a importância que a etapa de filtragem teve na geração de amostras com qualidade.

Os resultados para a base de dados *Heartbeat* podem ser vistos nos gráficos das Figuras 34 e 35. De acordo com os experimentos realizados, ficou claro a vantagem obtida pelo modelo utilizando a filtragem proposta. A melhoria na taxa de acerto da classe minoritária ocorreu durante todo o processo de inserção de

Figura 33 – Variação da precisão da classe minoritária na base *Skin Cancer*, à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem



amostras sintéticas. Observando o gráfico do AO sem a filtragem, é possível notar uma instabilidade no crescimento da precisão, principalmente após as primeiras amostras serem inseridas.

Figura 34 – Variação da precisão da classe minoritária 2 na base *Heartbeat*, à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.

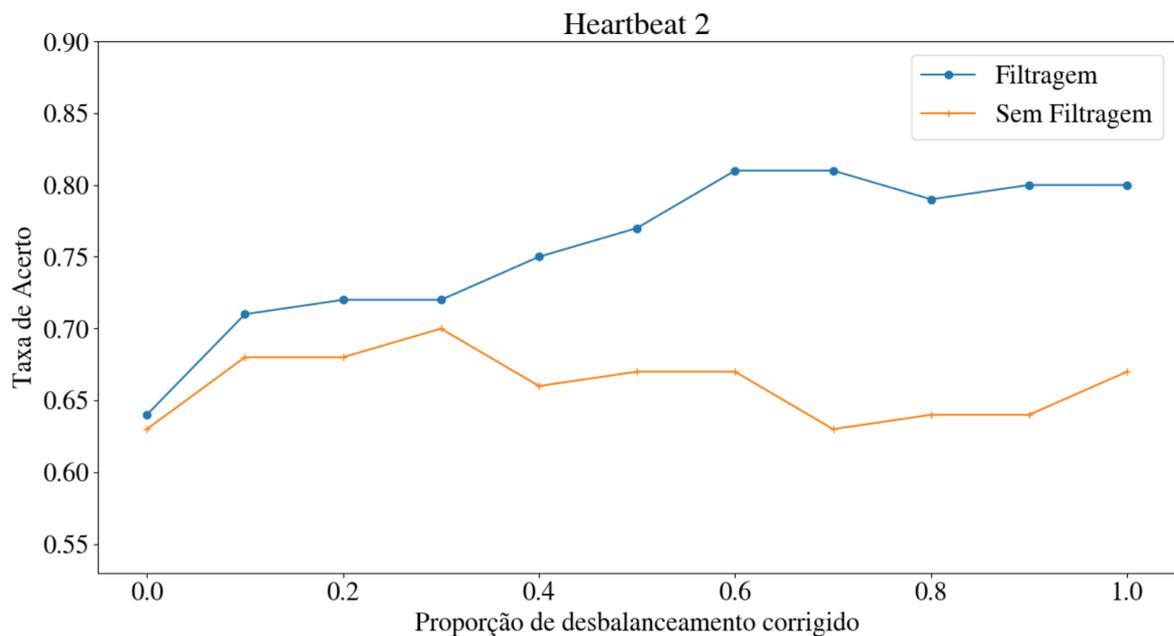
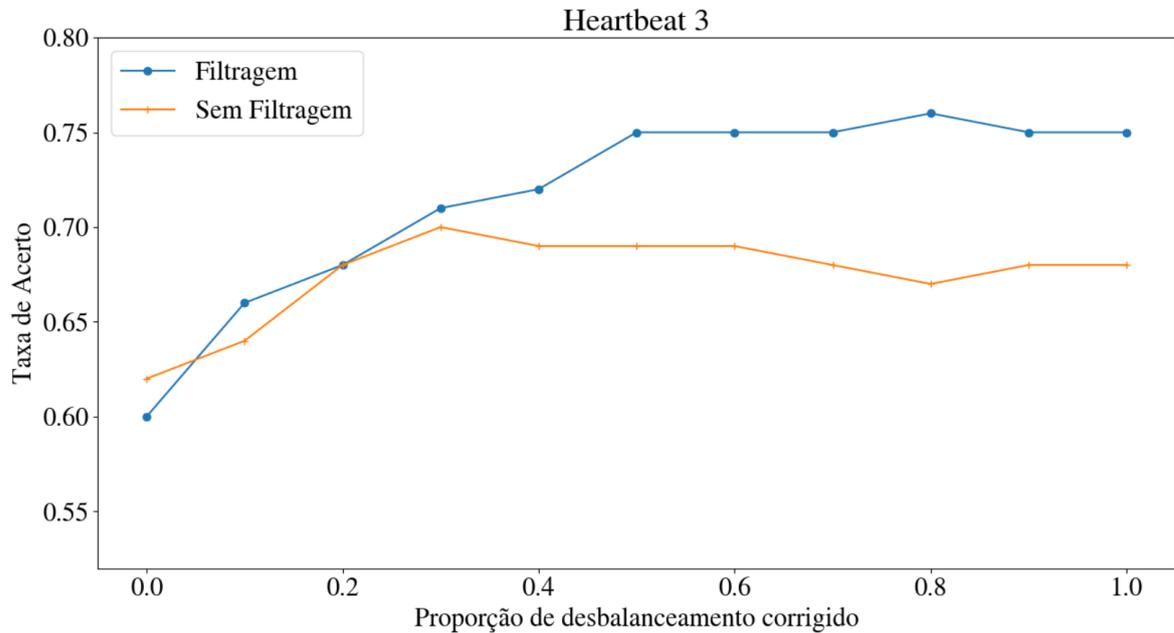
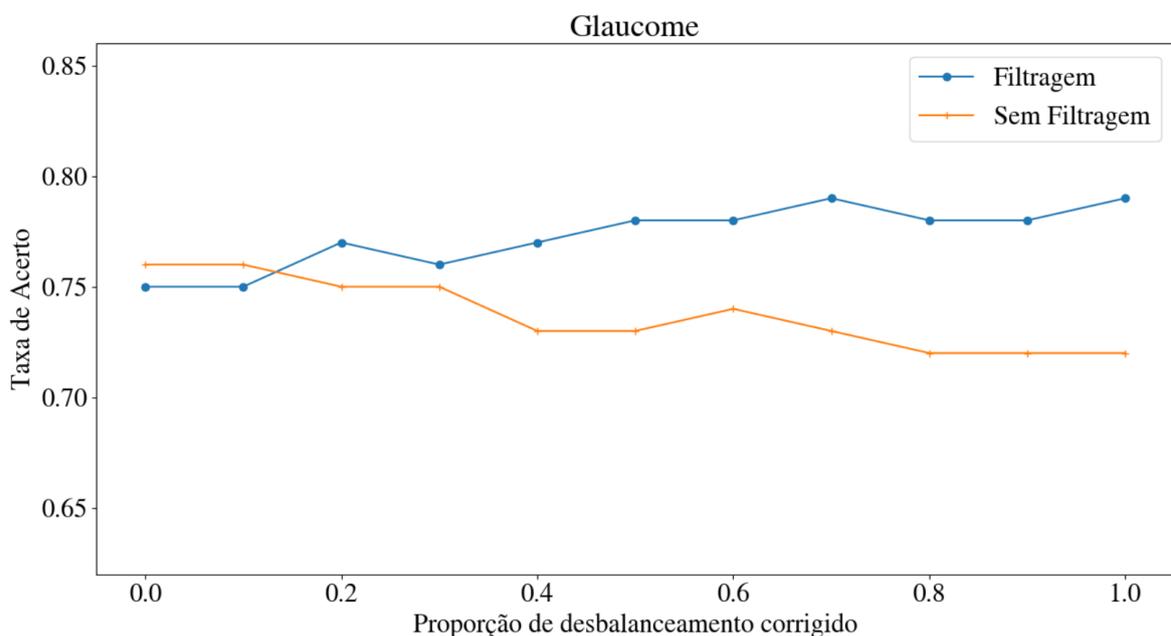


Figura 35 – Variação da precisão da classe minoritária 3 na base *Heartbeat*, à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.



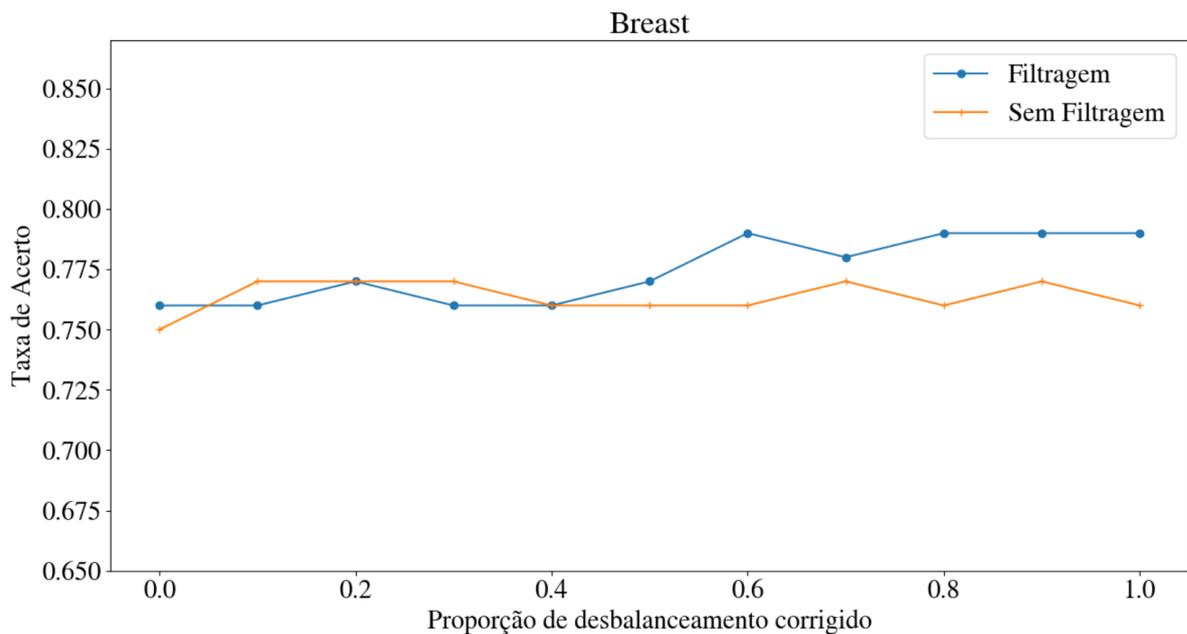
Os resultados para a base *Glaucome* podem ser vistos no gráfico da Figura 36. Apesar do método sem filtragem resultar numa precisão na classe minoritária maior nas primeiras 50 amostras, o AO com a filtragem passa superar o modelo sem filtragem a medida que mais amostras são geradas e inseridas,

Figura 36 – Variação da precisão da classe minoritária na base *Glaucome*, à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.



Observando agora os resultados para a base *Breast*, no gráfico da Figura 37, é possível notar uma equivalência nas taxas de acerto. Porém, após as primeiras 4000 amostras inseridas, ocorreu uma superioridade por parte do AO que utiliza a filtragem proposta. É observável que essa vantagem permaneceu até a inserção de todas as amostras sintéticas.

Figura 37 – Variação da precisão da classe minoritária na base *Breast*, à medida que são inseridas amostras sintéticas geradas pelo AO, com e sem filtragem.



Observando de forma generalizada, os resultados dos experimentos que avaliam a importância da etapa de filtragem proposta para o AO, nota-se que nas primeiras imagens há um aumento na precisão da classe minoritária. Porém, à medida que mais amostras são inseridas, o modelo que utiliza a filtragem proposta passa a ser superior no que diz respeito à contribuição na precisão. Sem a etapa de filtragem, há um crescimento na precisão nas primeiras amostras inseridas, mas com mais amostras inseridas aumentam as chances de amostras com baixa qualidade serem geradas e passarem a compor o *Oversampling*.

6.4 MÉTODOS TRADICIONAIS *vs* AO

Os resultados exibidos a seguir são relacionados ao experimento descrito na Seção 5.5.1, comparando o método proposto com os demais métodos de *Oversampling*, e são exibidos em formas de tabelas. Cada tabela contém uma medida referente à comparação de cada método em cada base de dados. Os resultados foram agru-

pados nas Seções 6.4.1, 6.4.2 e 6.4.3, contendo os resultados obtidos por cada classificador final, respectivamente: *VGG16*, *ResNet* e *DenseNet*.

6.4.1 Resultados utilizando o classificador VGG16

Inicialmente, utilizando o classificador VGG16, pode-se observar na Tabela 4 os resultados dos métodos considerando a média geométrica. O método proposto conseguiu superar os métodos tradicionais na maioria das bases de dados. Além disso, obteve um resultado superior às AC-GAN em todas as bases utilizadas para avaliação. Com relação aos métodos baseados em SMOTE, pode-se notar um bom desempenho na base *XRay*, com o próprio SVMSMOTE. Na base de dados *Breast*, pode-se observar que, mesmo o AO sendo superior aos demais, houveram resultados próximos com o SMOTE, AC-GAN e até mesmo o *Oversampling* aleatório do ROS.

Tabela 4 – Resultados da métrica G_{mean} nas bases de dados utilizando o classificador VGG16.

DATASET	ROS	SMOTE	ADASYN	BSMOTE	KSMOTE	SVMSMOTE	ACGAN	AO
Breast	0.81 (0.012)	0.81 (0.009)	0.78 (0.008)	0.77 (0.009)	0.77 (0.005)	0.78 (0.008)	0.75 (0.042)	0.82 (0.011)
Heartbeat	0.79 (0.012)	0.75 (0.021)	0.74 (0.021)	0.75 (0.017)	0.74 (0.021)	0.76 (0.013)	0.76 (0.036)	0.86 (0.016)
Glaucoma	0.80 (0.011)	0.80 (0.011)	0.79 (0.015)	0.75 (0.016)	0.74 (0.013)	0.76 (0.011)	0.74 (0.023)	0.83 (0.018)
XRay	0.75 (0.008)	0.78 (0.004)	0.80 (0.008)	0.79 (0.011)	0.78 (0.011)	0.81 (0.009)	0.76 (0.014)	0.80 (0.011)
Skin Cancer	0.70 (0.013)	0.70 (0.011)	0.71 (0.012)	0.71 (0.014)	0.71 (0.013)	0.72 (0.014)	0.74 (0.031)	0.78 (0.023)

A Tabela 5 traz os resultados utilizando a medida AUC-ROC. É natural que as taxas sejam maiores, pois a medida AUC-ROC é baseada em ranques. Porém, ainda é possível observar indícios da superioridade do AO, também com exceção apenas da base *XRay*, na qual há uma equivalência do método proposto com SVMSMOTE e BSMOTE.

Tabela 5 – Resultados da métrica AUC-ROC nas bases de dados utilizando o classificador VGG16.

DATASET	ROS	SMOTE	ADASYN	BSMOTE	KSMOTE	SVMSMOTE	ACGAN	AO
Breast	0.81 (0.010)	0.79 (0.011)	0.81 (0.013)	0.81 (0.012)	0.81 (0.012)	0.82 (0.014)	0.79 (0.020)	0.83 (0.014)
Heartbeat	0.82 (0.012)	0.80 (0.013)	0.81 (0.015)	0.81 (0.013)	0.81 (0.014)	0.82 (0.012)	0.82 (0.023)	0.87 (0.015)
Glaucoma	0.82 (0.014)	0.81 (0.015)	0.82 (0.013)	0.81 (0.011)	0.82 (0.013)	0.82 (0.013)	0.83 (0.031)	0.85 (0.016)
XRay	0.85 (0.013)	0.87 (0.013)	0.89 (0.015)	0.89 (0.013)	0.87 (0.011)	0.89 (0.015)	0.87 (0.021)	0.89 (0.014)
Skin Cancer	0.83 (0.008)	0.86 (0.010)	0.85 (0.011)	0.85 (0.010)	0.84 (0.010)	0.86 (0.009)	0.86 (0.023)	0.91 (0.011)

6.4.2 Resultados utilizando o classificador ResNet

A seguir, utilizando a CNN *ResNet* como classificador, pode-se ver na Tabela 6 os resultados com a média geométrica, G_{mean} , como medida. O método proposto foi

Tabela 6 – Resultados da métrica $Gmean$ nas bases de dados utilizando o classificador ResNet.

DATASET	ROS	SMOTE	ADASYN	BSMOTE	KSMOTE	SVMSMOTE	ACGAN	AO
Breast	0.78 (0.013)	0.80 (0.011)	0.80 (0.014)	0.79 (0.014)	0.77 (0.012)	0.81 (0.011)	0.79 (0.025)	0.82 (0.026)
Heartbeat	0.82 (0.020)	0.78 (0.018)	0.77 (0.018)	0.77 (0.019)	0.77 (0.019)	0.78 (0.017)	0.81 (0.026)	0.87 (0.015)
Glaucome	0.80 (0.014)	0.81 (0.013)	0.81 (0.012)	0.80 (0.013)	0.81 (0.015)	0.81 (0.015)	0.81 (0.023)	0.87 (0.018)
XRay	0.82 (0.012)	0.85 (0.012)	0.86 (0.014)	0.85 (0.015)	0.85 (0.015)	0.87 (0.013)	0.83 (0.024)	0.86 (0.017)
Skin Cancer	0.76 (0.014)	0.77 (0.013)	0.78 (0.017)	0.78 (0.016)	0.77 (0.014)	0.78 (0.015)	0.81 (0.032)	0.84 (0.015)

superior em quase todos as bases de dados, com a única exceção sendo a base de dados *XRay*, que resultou em um desempenho superior com o SVMSMOTE.

Considerando a medida de desempenho AUC-ROC, pode-se ver na Tabela 7 que a superioridade do método proposto foi mantida, porém há um equilíbrio considerável entre os métodos na base de dados *XRay*, notando uma equivalência entre o método SVMSMOTE e o método proposto AO.

Tabela 7 – Resultados da métrica AUC-ROC nas bases de dados utilizando o classificador ResNet.

DATASET	ROS	SMOTE	ADASYN	BSMOTE	KSMOTE	SVMSMOTE	ACGAN	AO
Breast	0.82 (0.012)	0.82 (0.012)	0.83 (0.013)	0.81 (0.014)	0.82 (0.011)	0.82 (0.012)	0.82 (0.022)	0.83 (0.014)
Heartbeat	0.80 (0.011)	0.79 (0.014)	0.80 (0.015)	0.80 (0.013)	0.81 (0.014)	0.82 (0.013)	0.81 (0.024)	0.87 (0.015)
Glaucome	0.81 (0.013)	0.82 (0.012)	0.82 (0.016)	0.81 (0.014)	0.80 (0.015)	0.81 (0.013)	0.82 (0.021)	0.85 (0.014)
XRay	0.84 (0.010)	0.87 (0.013)	0.86 (0.014)	0.86 (0.016)	0.87 (0.014)	0.88 (0.015)	0.86 (0.025)	0.88 (0.012)
Skin Cancer	0.82 (0.014)	0.81 (0.013)	0.81 (0.012)	0.82 (0.015)	0.83 (0.013)	0.83 (0.014)	0.84 (0.019)	0.89 (0.016)

6.4.3 Resultados utilizando o classificador *DenseNet*

Por fim, utilizando a rede *DenseNet* como classificador, é possível perceber na Tabela 8 os resultados considerando a média geométrica. O método proposto foi superior aos demais na maioria das bases de dados. A única exceção é a base de dados *XRay*, na qual o SVMSMOTE obteve um resultado equivalente à proposta.

Tabela 8 – Resultados da métrica $Gmean$ nas bases de dados utilizando o classificador *DenseNet*.

DATASET	ROS	SMOTE	ADASYN	BSMOTE	KSMOTE	SVMSMOTE	ACGAN	AO
Breast	0.74 (0.022)	0.74 (0.022)	0.73 (0.021)	0.77 (0.020)	0.74 (0.020)	0.76 (0.019)	0.77 (0.027)	0.78 (0.020)
Heartbeat	0.76 (0.023)	0.75 (0.022)	0.75 (0.021)	0.76 (0.022)	0.73 (0.021)	0.75 (0.023)	0.76 (0.025)	0.84 (0.021)
Glaucome	0.74 (0.014)	0.75 (0.014)	0.74 (0.015)	0.75 (0.014)	0.75 (0.016)	0.76 (0.015)	0.75 (0.021)	0.79 (0.017)
XRay	0.80 (0.013)	0.82 (0.014)	0.82 (0.014)	0.83 (0.017)	0.83 (0.018)	0.84 (0.012)	0.82 (0.023)	0.84 (0.018)
Skin Cancer	0.74 (0.013)	0.74 (0.016)	0.75 (0.017)	0.77 (0.016)	0.76 (0.017)	0.76 (0.016)	0.79 (0.024)	0.83 (0.019)

Os resultados utilizando a *DenseNet* como classificador são exibidos Tabela 9, os quais estão utilizando a AUC-ROC como medida. Também é percebido que o método proposto foi superior aos métodos utilizando SMOTE e também às AC-GAN. A única exceção é foi observada também na base de dados *XRay*, na qual o SVMS-

MOTE, juntamente com o BSMOTE, apresentaram resultados equivalentes ao método proposto, AO.

Tabela 9 – Resultados da métrica AUC-ROC nas bases de dados utilizando o classificador *DenseNet*.

DATASET	ROS	SMOTE	ADASYN	BSMOTE	KSMOTE	SVMSMOTE	ACGAN	AO
Breast	0.77 (0.024)	0.75 (0.022)	0.75 (0.023)	0.76 (0.024)	0.76 (0.024)	0.77 (0.023)	0.76 (0.026)	0.80 (0.026)
Heartbeat	0.80 (0.023)	0.78 (0.022)	0.78 (0.023)	0.77 (0.021)	0.77 (0.021)	0.79 (0.018)	0.80 (0.024)	0.86 (0.023)
Glaucome	0.81 (0.018)	0.80 (0.018)	0.81 (0.018)	0.82 (0.022)	0.81 (0.019)	0.82 (0.018)	0.81 (0.021)	0.85 (0.022)
XRay	0.80 (0.014)	0.85 (0.015)	0.84 (0.016)	0.86 (0.015)	0.83 (0.013)	0.86 (0.013)	0.83 (0.025)	0.86 (0.016)
Skin Cancer	0.84 (0.019)	0.83 (0.019)	0.83 (0.021)	0.83 (0.021)	0.82 (0.019)	0.83 (0.020)	0.84 (0.023)	0.88 (0.020)

6.4.4 Análise dos Resultados

De uma forma sumarizada, pode-se observar alguns fatores interessantes com relação aos resultados do experimento de comparação dos métodos de *Oversampling*. Primeiramente, foi observado uma certa dificuldade em determinar, dentre métodos tradicionais, baseados em SMOTE, qual obteve o melhor resultado geral. Em segundo lugar, pode-se observar que o *Oversampling* através das AC-GAN obteve um resultado médio um pouco melhor que alguns métodos baseados em SMOTE, porém foi observado uma instabilidade maior, refletida pelo aumento no desvio padrão. Isso pode ter ocorrido pelo fator aleatório adicionado ao treinamento das AC-GAN, que podem ter afetado a convergência do modelo e, conseqüentemente, a qualidade das amostras geradas. Por fim, observando os resultados gerais do método proposto AO, pode-se notar indícios de uma superioridade na maioria dos casos, mesmo se considerado o desvio padrão, o qual foi um pouco maior que os métodos tradicionais.

7 CONCLUSÕES

O problema de aprendizagem envolvendo bases de dados desbalanceadas há muito tempo vem sendo fonte de investigação e de pesquisa elaboração de métodos que possam resolvê-lo ou ao menos reduzir os efeitos negativos. Esse problema é importante pois uma série de classificadores podem se tornar ineficazes quando são treinados sobre bases de dados desbalanceadas. Isso porque esses classificadores podem ficar especializados somente nas classes com maior quantidade de amostras, fornecendo um desempenho insuficiente nas classes minoritárias. Como nem sempre é possível a coleta de novas amostras, os métodos chamados de *Oversampling* vêm sendo estudados para geração de amostras a partir de amostras existentes. Com base na *Generative Adversarial Networks* (GANs), a qual compõe o estado da arte para geração de amostras, o presente trabalho propôs o chamado *Adversarial Oversampling* - AO. Esse método utilizou-se da *Auxiliary Classifier GANs* (AC-GAN) como rede adversária e base do método, para aprender a distribuição das classes na base de dados e gerar amostras para a classe minoritária, até que haja o equilíbrio das classes. Durante o treinamento do AO, é dada uma maior atenção às amostras consideradas difíceis, estratégia característica dos alguns métodos *Oversampling*. Com isso, o AO tenta gerar amostras úteis, que consigam ajudar o classificador a construir uma melhor região de fronteira. As amostras sintéticas geradas ainda passam por uma filtragem utilizando o próprio discriminador **D** do AO, para que haja uma tendência a serem selecionadas apenas amostras realistas e que pertencem a classe minoritária, diminuindo a inserção de amostras ruidosas na base de dados durante a realização do *Oversampling* na classe minoritária.

Para avaliar a viabilidade da proposta como método de *Oversampling*, foram feitas algumas análises. Primeiramente, foi analisado o aspecto visual das imagens geradas pelo AO, pelas quais foi possível observar uma maior qualidade, pois as características visuais foram mais fieis às amostras originais, do que as imagens geradas por interpolação e sem aprendizado, como as técnicas baseadas em SMOTE comparadas. Uma outra análise foi feita observando também o quão a precisão do classificador na classe minoritária cresce à medida que são inseridas amostras geradas pelo AO. Os resultados dessa análise dão indícios que as amostras geradas contribuíram para o crescimento da precisão do classificador na classe minoritária. Além disso, foi visto que nem sempre quanto mais amostras inseridas, maior é essa precisão. Também foi realizada uma análise do quão a etapa de filtragem pode ser importante para o método proposto. A partir dessa análise foram observados

indícios que o AO com a filtragem proposta melhorou a qualidade das amostras inseridas durante o *Oversampling*. Por fim, o AO foi comparado com outros métodos de *Oversampling* observando o quão as amostras geradas para a classe minoritária contribuíram para o acerto final do classificador. Para isso, foram realizados experimentos comparativos em seis bases de dados de imagens. Os resultados deram indícios de que o AO obteve uma certa superioridade na maioria das bases. As exceções foram observadas na base *Breast* e na base de dados *XRay*, nas quais se observou um equilíbrio considerável no acerto médio dos métodos comparados.

7.1 TRABALHOS FUTUROS

De fato, houveram indícios que o AO, utilizando do poder das GANs para geração de amostras sintéticas, pode ser uma alternativa como método de *Oversampling* para balancear bases de dados envolvendo imagens. Porém, o trabalho trás um estudo inicial, ainda há necessidade de esforços para confirmar e avaliar os resultados obtidos, além de estudos visando o aperfeiçoamento do AO como método de *Oversampling*. Para isso, os seguintes trabalhos futuros são oportunidades de pesquisa:

- Realizar mais avaliações para garantir a qualidade das amostras geradas pelo método proposto. Uma possibilidade é aumentar a influência das amostras sintéticas no experimento, utilizando somente as amostras geradas para compor as bases de treinamento dos classificadores.
- Realizar mais experimentos para aumentar mais a validação do *Adversarial Oversampling* (AO), pode-se compará-lo a mais métodos de balanceamento, considerando também *Undersampling* e a utilização *ensemble*. Além disso, pode-se compará-lo com métodos mais recentes, incluindo métodos que já façam uso de GANs para *Oversampling*.
- O AO é modular e simplifica a aplicação em um domínio diferente de imagens. Isso é possível alterando a rede adversária para uma relacionada ao tipo de dado que se deseja gerar. Por exemplo, uma GANs recorrente (Arnelid; Zec; Mohammadiha, 2019), a base de *LSTMs*, para balancear bases de dados de classificação de séries temporais.
- Uso de GANs mais recentes como Rede Adversária, o que pode melhorar a estabilidade e a qualidade das amostras geradas pelo AO.

REFERÊNCIAS

- Abdoh, S. F.; Abo Rizka, M.; Maghraby, F. A. Cervical cancer diagnosis using random forest classifier with smote and feature reduction techniques. *IEEE Access*, v. 6, p. 59475–59485, 2018. ISSN 2169-3536.
- Arnelid, H.; Zec, E. L.; Mohammadiha, N. Recurrent conditional generative adversarial networks for autonomous driving sensor modelling. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. [S.l.: s.n.], 2019. p. 1613–1618.
- BARANDELA, R.; SÁNCHEZ, J.; GARCÍA, V.; RANGEL, E. Strategies for learning in class imbalance problems. *Pattern Recognition*, v. 36, n. 3, p. 849 – 851, 2003. ISSN 0031-3203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0031320302002571>>.
- BATISTA, G. E. A. P. A.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, Association for Computing Machinery, New York, NY, USA, v. 6, n. 1, p. 20–29, jun. 2004. ISSN 1931-0145. Disponível em: <<https://doi.org/10.1145/1007730.1007735>>.
- BHATTARAI, B.; BAEK, S.; BODUR, R.; KIM, T.-K. *Sampling Strategies for GAN Synthetic Data*. 2019.
- BOZORGTABAR, B.; MAHAPATRA, D.; TENG, H. von; POLLINGER, A.; EBNER, L.; THIRAN, J.-P.; REYES, M. Informative sample generation using class aware generative adversarial networks for classification of chest xrays. *Computer Vision and Image Understanding*, v. 184, p. 57 – 65, 2019. ISSN 1077-3142. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S107731421930061X>>.
- BRADLEY, A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, v. 30, n. 7, p. 1145 – 1159, 1997. ISSN 0031-3203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0031320396001422>>.
- CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, AI Access Foundation, USA, v. 16, n. 1, p. 321–357, jun. 2002. ISSN 1076-9757. Disponível em: <<http://dl.acm.org/citation.cfm?id=1622407.1622416>>.
- CRUZ-ROA, A.; BASAVANHALLY, A.; GONZÁLEZ, F.; GILMORE, H.; FELDMAN, M.; GANESAN, S.; SHIH, N.; TOMASZEWSKI, J.; MADABHUSHI, A. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. *Progress in Biomedical Optics and Imaging - Proceedings of SPIE*, v. 9041, 02 2014.
- Demidova, L.; Klyueva, I. Svm classification: Optimization with the smote algorithm for the class imbalance problem. In: *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. [S.l.: s.n.], 2017. p. 1–4. ISSN null.

- DEVLIN, J.; CHANG, M.; LEE, K.; TOUTANOVA, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. Disponível em: <<http://arxiv.org/abs/1810.04805>>.
- GECER, B.; BHATTARAI, B.; KITTLER, J.; KIM, T. Semi-supervised adversarial learning to generate photorealistic face images of new identities from 3d morphable model. *CoRR*, abs/1804.03675, 2018. Disponível em: <<http://arxiv.org/abs/1804.03675>>.
- Ghosh, K.; Banerjee, A.; Chatterjee, S.; Sen, S. Imbalanced twitter sentiment analysis using minority oversampling. In: *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*. [S.l.: s.n.], 2019. p. 1-5. ISSN 2325-5986.
- GOODFELLOW, I. J.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. *Generative Adversarial Networks*. 2014.
- Haibo He; Yang Bai; Garcia, E. A.; Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. [S.l.: s.n.], 2008. p. 1322-1328. ISSN 2161-4393.
- HAN, H.; WANG, W.-Y.; MAO, B.-H. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: HUANG, D.-S.; ZHANG, X.-P.; HUANG, G.-B. (Ed.). *Advances in Intelligent Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 878-887. ISBN 978-3-540-31902-3.
- He, H.; Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, v. 21, n. 9, p. 1263-1284, Sep. 2009. ISSN 2326-3865.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>>.
- Hearst, M. A.; Dumais, S. T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intelligent Systems and their Applications*, v. 13, n. 4, p. 18-28, July 1998. ISSN 2374-9423.
- HUANG, G.; LIU, Z.; WEINBERGER, K. Q. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. Disponível em: <<http://arxiv.org/abs/1608.06993>>.
- IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. JMLR.org, 2015. (ICML'15), p. 448-456. Disponível em: <<http://dl.acm.org/citation.cfm?id=3045118.3045167>>.
- JIN, Y.; ZHANG, J.; LI, M.; TIAN, Y.; ZHU, H.; FANG, Z. Towards the automatic anime characters creation with generative adversarial networks. *CoRR*, abs/1708.05509, 2017. Disponível em: <<http://arxiv.org/abs/1708.05509>>.

JOHNSON, J. M.; KHOSHGOFTAAR, T. M. Survey on deep learning with class imbalance. *Journal of Big Data*, v. 6, n. 1, p. 27, 2019. ISSN 2196-1115. Disponível em: <<https://doi.org/10.1186/s40537-019-0192-5>>.

KARRAS, T.; AILA, T.; LAINE, S.; LEHTINEN, J. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. Disponível em: <<http://arxiv.org/abs/1710.10196>>.

KERMANY, D. S.; ZHANG, K.; GOLDBAUM, M. H. Labeled optical coherence tomography (oct) and chest x-ray images for classification. In: . [S.l.: s.n.], 2018.

KINGMA, D.; BA, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

KRAWCZYK, B. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, v. 5, n. 4, p. 221–232, 2016. ISSN 2192-6360. Disponível em: <<https://doi.org/10.1007/s13748-016-0094-0>>.

LAST, F.; DOUZAS, G.; BAÇÃO, F. Oversampling for imbalanced learning based on k-means and SMOTE. *CoRR*, abs/1711.00837, 2017. Disponível em: <<http://arxiv.org/abs/1711.00837>>.

LECUN, Y.; HAFFNER, P.; BOTTOU, L.; BENGIO, Y. Object recognition with gradient-based learning. In: *Shape, Contour and Grouping in Computer Vision*. Berlin, Heidelberg: Springer-Verlag, 1999. p. 319. ISBN 3540667229.

LEMAÎTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, v. 18, n. 17, p. 1–5, 2017. Disponível em: <<http://jmlr.org/papers/v18/16-365.html>>.

Li, J.; He, H.; Li, L. Cgan-mbl for reliability assessment with imbalanced transmission gear data. *IEEE Transactions on Instrumentation and Measurement*, v. 68, n. 9, p. 3173–3183, Sep. 2019. ISSN 1557-9662.

Li, Z.; Jin, Y.; Li, Y.; Lin, Z.; Wang, S. Imbalanced adversarial learning for weather image generation and classification. In: *2018 14th IEEE International Conference on Signal Processing (ICSP)*. [S.l.: s.n.], 2018. p. 1093–1097. ISSN 2164-5221.

Lima, J. L. P.; Macêdo, D.; Zanchettin, C. Heartbeat anomaly detection using adversarial oversampling. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2019. p. 1–7. ISSN 2161-4393.

LIU, M.-Y.; TUZEL, O. *Coupled Generative Adversarial Networks*. 2016.

LIU, W.; LUO, Z.; LI, S. Improving deep ensemble vehicle classification by using selected adversarial samples. *Knowledge-Based Systems*, v. 160, p. 167 – 175, 2018. ISSN 0950-7051. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950705118303216>>.

LIU, Z.; LIU, Q. Studying cost-sensitive learning for multi-class imbalance in internet traffic classification. *The Journal of China Universities of Posts and Telecommunications*, v. 19, n. 6, p. 63 – 72, 2012. ISSN 1005-8885. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1005888511603191>>.

Lloyd, S. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, v. 28, n. 2, p. 129–137, March 1982. ISSN 1557-9654.

LU, D.; WENG, Q. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, v. 28, p. 823 – 870, 03 2007.

MA, L.; JIA, X.; SUN, Q.; SCHIELE, B.; TUYTELAARS, T.; GOOL, L. V. Pose guided person image generation. *CoRR*, abs/1705.09368, 2017. Disponível em: <<http://arxiv.org/abs/1705.09368>>.

MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. [S.l.: s.n.], 2013.

MACÊDO, D.; REN, T. I.; ZANCHETTIN, C.; OLIVEIRA, A. L. I.; TAPP, A.; LUDERMIR, T. *Distinction Maximization Loss: Fast, Scalable, Turnkey, and Native Neural Networks Out-of-Distribution Detection simply by Replacing the SoftMax Loss*. 2019.

Mathew, J.; Pang, C. K.; Luo, M.; Leong, W. H. Classification of imbalanced data by oversampling in kernel space of support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, v. 29, n. 9, p. 4065–4076, Sep. 2018. ISSN 2162-2388.

MIRZA, M.; OSINDERO, S. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. Disponível em: <<http://arxiv.org/abs/1411.1784>>.

Moody, G. B.; Mark, R. G. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, v. 20, n. 3, p. 45–50, May 2001. ISSN 1937-4186.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. USA: Omnipress, 2010. (ICML'10), p. 807–814. ISBN 978-1-60558-907-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=3104322.3104425>>.

Nath, S. S.; Mishra, G.; Kar, J.; Chakraborty, S.; Dey, N. A survey of image classification methods and techniques. In: *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. [S.l.: s.n.], 2014. p. 554–557. ISSN null.

NEELAKANTAN, A.; VILNIS, L.; LE, Q. V.; SUTSKEVER, I.; KAISER, L.; KURACH, K.; MARTENS, J. *Adding Gradient Noise Improves Learning for Very Deep Networks*. 2015.

NGUYEN, H. M.; COOPER, E. W.; KAMEI, K. Borderline oversampling for imbalanced data classification. *Int. J. Knowl. Eng. Soft Data Paradigm.*, Inderscience Publishers, Inderscience Publishers, Geneva, SWITZERLAND, v. 3, n. 1, p. 4–21, abr. 2011. ISSN 1755-3210. Disponível em: <<http://dx.doi.org/10.1504/IJKESDP.2011.039875>>.

ODENA, A.; OLAH, C.; SHLENS, J. *Conditional Image Synthesis With Auxiliary Classifier GANs*. 2016.

RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, v. 29, p. 1–98, 06 2017.

REED, S. E.; AKATA, Z.; YAN, X.; LOGESWARAN, L.; SCHIELE, B.; LEE, H. Generative adversarial text to image synthesis. *CoRR*, abs/1605.05396, 2016. Disponível em: <<http://arxiv.org/abs/1605.05396>>.

Rezaei, M.; Yang, H.; Meinel, C. Generative adversarial framework for learning multiple clinical tasks. In: *2018 Digital Image Computing: Techniques and Applications (DICTA)*. [S.l.: s.n.], 2018. p. 1–8. ISSN null.

REZAEI, M.; YANG, H.; MEINEL, C. voxel-gan: Adversarial framework for learning imbalanced brain tumor segmentation. In: CRIMI, A.; BAKAS, S.; KUIJF, H.; KEYVAN, F.; REYES, M.; WALSUM, T. van (Ed.). *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Cham: Springer International Publishing, 2019. p. 321–333. ISBN 978-3-030-11726-9.

Saad, A. I.; Omar, Y. M. K.; Maghraby, F. A. Predicting drug interaction with adenosine receptors using machine learning and smote techniques. *IEEE Access*, v. 7, p. 146953–146963, 2019. ISSN 2169-3536.

Salehinejad, H.; Valaee, S.; Dowdell, T.; Colak, E.; Barfett, J. Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2018. p. 990–994. ISSN 2379-190X.

SALIMANS, T.; GOODFELLOW, I.; ZAREMBA, W.; CHEUNG, V.; RADFORD, A.; CHEN, X. *Improved Techniques for Training GANs*. 2016.

Satriaji, W.; Kusumaningrum, R. Effect of synthetic minority oversampling technique (smote), feature representation, and classification algorithm on imbalanced sentiment analysis. In: *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*. [S.l.: s.n.], 2018. p. 1–5. ISSN null.

SEZER, O. B.; GUDELEK, M. U.; OZBAYOGLU, A. M. *Financial Time Series Forecasting with Deep Learning : A Systematic Literature Review: 2005-2019*. 2019.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

Sornam, M.; Muthusubash, K.; Vanitha, V. A survey on image classification and activity recognition using deep convolutional neural network architecture. In: *2017 Ninth International Conference on Advanced Computing (ICoAC)*. [S.l.: s.n.], 2017. p. 121–126. ISSN null.

- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUT-DINOV, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, p. 1929–1958, 2014. Disponível em: <<http://jmlr.org/papers/v15/srivastava14a.html>>.
- Sultana, F.; Sufian, A.; Dutta, P. Advancements in image classification using convolutional neural network. In: *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. [S.l.: s.n.], 2018. p. 122–129.
- SØNDERBY, C. K.; CABALLERO, J.; THEIS, L.; SHI, W.; HUSZÁR, F. *Amortised MAP Inference for Image Super-resolution*. 2016.
- TAIGMAN, Y.; POLYAK, A.; WOLF, L. *Unsupervised Cross-Domain Image Generation*. 2016.
- TIAN, D. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, v. 8, p. 385–395, 01 2013.
- VU, L.; BUI, C. T.; NGUYEN, Q. U. A deep learning based method for handling imbalanced problem in network traffic classification. In: *Proceedings of the Eighth International Symposium on Information and Communication Technology*. New York, NY, USA: Association for Computing Machinery, 2017. (SoICT 2017), p. 333–339. ISBN 9781450353281. Disponível em: <<https://doi.org/10.1145/3155133.3155175>>.
- Wang, Q.; Zhou, X.; Wang, C.; Liu, Z.; Huang, J.; Zhou, Y.; Li, C.; Zhuang, H.; Cheng, J. Wgan-based synthetic minority over-sampling technique: Improving semantic fine-grained classification for lung nodules in ct images. *IEEE Access*, v. 7, p. 18450–18463, 2019. ISSN 2169-3536.
- YANMINSUN; WONG, A.; KAMEL, M. S. Classification of imbalanced data: a review. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 23, 11 2011.
- Zhang, C.; Guo, J.; Lu, J. Research on classification method of high-dimensional class-imbalanced data sets based on svm. In: *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*. [S.l.: s.n.], 2017. p. 60–67. ISSN null.
- ZHANG, H.; XU, T.; LI, H.; ZHANG, S.; WANG, X.; HUANG, X.; METAXAS, D. *StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks*. 2016.
- ZHANG, Z.; YIN, F.; LIU, J.; WONG, W.; TAN, N.; LEE, B.-H.; CHENG, J.; WONG, T.-Y. Origa(-light): An online retinal fundus image database for glaucoma analysis and research. In: . [S.l.: s.n.], 2010. v. 2010, p. 3065–8.
- ZHU, J.; PARK, T.; ISOLA, P.; EFROS, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. Disponível em: <<http://arxiv.org/abs/1703.10593>>.