Penggunaan Feature Space SMOTE Untuk Mengurangi Overfitting Akibat Imbalance Dataset

Utilization of Feature Space SMOTE to Reduce Overfitting Due to Imbalanced Dataset

Wira Adi Kurniawan[1], Abu Salam[2]
[1,2]Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
E-mail: [1]111202012966@mhs.dinus.ac.id, [2]abu.salam@dsn.dinus.ac.id

Abstrak

Pembuatan model klasifikasi memerlukan beberapa hal yang penting untuk diperhatikan demi mendapatkan model yang memiliki performa terbaik. Indikator suatu model disebut baik dapat dilihat salah satunya dari tingginya nilai akurasi dan *f1-score* yang dihasilkan dari model tersebut. Rendahnya nilai *loss* juga merupakan salah satu indikator model tersebut memiliki performa yang baik. Untuk dapat membuat model yang baik, diperlukan beberapa syarat seperti arsitektur yang tepat dan data yang berkualitas. Pemilihan model yang terlalu sederhana akan mengakibatkan model memiliki performa yang buruk, begitupun jika model terlalu kompleks tidak akan menghasilkan performa yang baik pula, oleh karena itu model yang dipilih haruslah model yang tepat dan sesuai dengan jenis data yang digunakan. Data yang berkualitas juga merupakan faktor penting agar model memiliki performa maksimal. Data dapat dikatakan berkualitas jika memenuhi syarat seperti jumlahnya cukup, distribusi datanya seimbang tiap kelas, memiliki keanekaragaman dan memiliki kebersihan yang baik. Pada penelitian ini, dilakukan pembuatan model klasifikasi CT *Kidney Stone* dengan dataset yang *imbalance*. Dataset diperoleh dari sumber publik yaitu Kaggle. Pembuatan model menggunakan algoritma CNN karena CNN merupakan salah satu algoritma yang terbaik dalam membuat klasifikasi gambar. Pembuatan model menggunakan 3 cara untuk melihat model yang memiliki performa paling baik. Model pertama dibuat dengan data *train* yang *imbalance*. Model kedua dibuat dengan melakukan augmentasi data untuk menambah keragaman data. Model ketiga dibuat dengan SMOTE *oversampling* untuk menyeimbangkan distribusi data. Setelah itu ketiga model tersebut akan diuji dengan menggunakan data privat untuk melihat performa pengujian dan melihat tingkat *overfitting* yang terjadi. Penelitian ini menghasilkan bahwa model yang memiliki performa terbaik adalah model ketiga yang menggunakan SMOTE.

Kata kunci: CNN, imbalance, smote, augmentation, overfitting

Abstract

The creation of a classification model requires several important considerations to ensure the best performance. The effectiveness of a model can be indicated by high accuracy and F1-score values generated by the model. A low loss value is also an indicator of a well-performing model. To create a good model, several requirements such as the right architecture and quality data are needed. Selecting a model that is too simple will result in poor performance, and similarly, if the model is too complex, it will not yield good performance either. Therefore, the chosen model must be appropriate and suitable for the type of data used. Quality data is also a crucial factor for maximizing model performance. Data can be considered of high quality if it meets criteria such as sufficient quantity, balanced distribution across classes, diversity, and cleanliness. In this study, a CT Kidney Stone classification model was created using an imbalanced dataset obtained from a public source, Kaggle. The model was built using CNN algorithm as CNN is one of the best algorithms for image classification. The model creation process involved three approaches to determine the best-performing model. The first model was built using the imbalanced training data. The second model was created by augmenting the data to increase data diversity. The third model was built using SMOTE oversampling to balance the data distribution. Subsequently, these three models were tested using private data to assess their testing performance and identify the

level of overfitting. The research findings indicate that the model with the best performance is the third model utilizing SMOTE

Keywords: CNN, imbalance, smote, augmentation, overfitting

1. PENDAHULUAN

Deep Learning merupakan evolusi dari jaringan syaraf tiruan menuju arsitektur jaringan syaraf yang semakin mendalam dengan peningkatan kemampuan belajar [1]. *Deep Learning* banyak dipakai untuk pengenalan gambar karena memiliki kemampuan yang baik mempelajari pola pada gambar. Arsitektur utama *deep learning* untuk pemrosesan gambar yaitu *Convolutional Neural Network* (CNN) [2]. CNN adalah jenis jaringan syaraf tertentu yang dirancang untuk memproses data yang memiliki struktur seperti grid [3]. Arsitektur CNN banyak dipakai untuk mengenali dan mengklasifikasi berbagai gambar seperti: buah, MRI, USG, ataupun CT-Scan. CT-Scan adalah salah satu skema paling umum untuk mendeteksi penyakit dalam praktik klinis [4].

Untuk dapat membuat klasifikasi data gambar menjadi kelas-kelas tertentu, perlu dibuat teknik pengklasifikasian yang tepat agar memiliki hasil yang akurat dan tingkat kesalahannya rendah. Beberapa hal yang perlu diperhatikan dalam membuat model pengklasifikasian seperti: jumlah data, kompleksitas model, keseimbangan data, dan sebagainya. Keseimbangan data adalah hal yang penting, jika jumlah data pada satu kelas memiliki jumlah lebih banyak dan pada kelas lain terdapat data yang lebih sedikit maka data tersebut bisa dikatakan tidak seimbang atau *imbalance* [5]. Hal-hal tersebut dapat menjadi penyebab suatu model memiliki akurasi yang buruk bahkan mengalami *overfitting*. *Overfitting* adalah kondisi dimana model memiliki performa yang sangat bagus pada saat *training*, tetapi pada saat *testing* menghasilkan performa yang buruk [6]. Hal ini dapat terjadi karena model cenderung menghafal data *train*, termasuk noise yang tidak penting pada data *train* tersebut. Seharusnya model mempelajari pola dari data *train* tersebut bukan menghafalnya. Untuk mengatasi masalah distribusi kelas tersebut perlu dilakukan *oversampling* atau *undersampling*. *Oversampling* yaitu suatu cara menyeimbangkan distribusi kelas dengan cara menyintesis kelas minoritas agar seimbang jumlahnya dengan kelas mayoritas [7], [8]. *Oversampling* biasanya menyintesis kelas baru berdasarkan tetangga terdekat *(nearest neighbors)* dari kelas minoritas. *Oversampling* memiliki berbagai metode, contohnya yaitu SMOTE, ADASYN, *Random Oversampling* (ROS), dan masih banyak pengembangan metode oversampling lainnya. SMOTE adalah metode yang paling banyak digunakan dalam menyeimbangkan distribusi data antar kelas. Ide pokok di balik SMOTE adalah mengambil secara acak sampel titik data minoritas buatan sepanjang segmen garis yang menghubungkan titik data minoritas di antara *k* dari tetangga terdekat kelas minoritas [9]. SMOTE merupakan contoh dari *oversampling* yang dilakukan pada *feature-space*. Bukan hanya pada *feature-space*, *oversampling* juga pada dilakukan pada *input-space*.

Secara umum, *oversampling* ataupun *undersampling* dapat dibagi menjadi dua kategori, *yaitu input-space* dan *feature-space* [10]. Pada *input-space oversampling*, dilakukan manipulasi data asli dari kelas minoritas sebelum adanya ekstraksi fitur apapun. Prinsip kerja *input-space oversampling* yaitu memperbanyak data dengan proses tranformasi dan augmentasi langsung pada data itu sendiri sehingga tercipta data baru yang lebih beragam. Dengan data yang lebih beragam, diharapkan pelatihan model dapat memberikan hasil yang lebih baik. Contoh proses augmentasi yaitu rotasi, *flip*, dan *zoom*. Sedangkan *feature-space* merupakan ruang yang digunakan untuk melakukan ekstraksi fitur. *Feature-space oversampling* yaitu pendekatan untuk menangani *imbalance class* pada dataset. *Feature-space* ada bermacam tekniknya, yang paling umum dipakai yaitu SMOTE. SMOTE menciptakan sampel sintetis yang berasal dari tetangga terdekat.

Beberapa penelitian telah dilakukan oleh peneliti sebelumnya dan memiliki topik yang serupa dengan penelitian yang akan dilakukan. Contohnya yaitu penelitian [11]. Penelitian tersebut membandingkan klasifikasi berita dengan data yang *imbalance* dan klasifikasi dengan data yang sudah dilakukan *oversampling* dengan SMOTE. Penelitian tersebut menggunakan 200

data, dengan kelas mayoritas berjumlah 176, dan kelas minoritas berjumlah 24. Teknik pengklasifikasian pada penelitian tersebut menggunakan algoritma KNN. Hasil penelitian tersebut menemukan bahwa penggunaan *oversampling* dengan metode SMOTE efektif untuk menaikkan performa model dengan rata-rata peningkatan akurasi sebesar 3,36. Contoh penelitian yang lain yaitu [12]. Pada penelitian tersebut, data terdiri dari dua kelas dengan kelas 0 berjumlah 210 dan kelas 1 berjumlah 40. Untuk menyeimbangkan distribusi data, dilakukan *oversampling* menggunakan SMOTE dengan memanfaatkan *library* imb-learn. Dalam penelitian tersebut, digunakan 4 macam algoritma untuk membuat model, yaitu algoritma SVM, Naïve Bayes, KNN, dan Decision Tree. Hasilnya, 3 dari 4 algoritma menghasilkan model yang memiliki performa lebih baik dengan menambahkan *oversampling* SMOTE. Penggunaan SMOTE meningkatkan akurasi pada algoritma Naïve Bayes sebesar 24%, pada algoritma KNN sebesar 1%, dan pada algoritma Decision Tree sebesar 2%. Penelitian [13] juga menyimpulkan bahwa teknik oversampling SMOTE dapat meningkatkan akurasi model yang dibuat, bahkan hingga 7,40% .

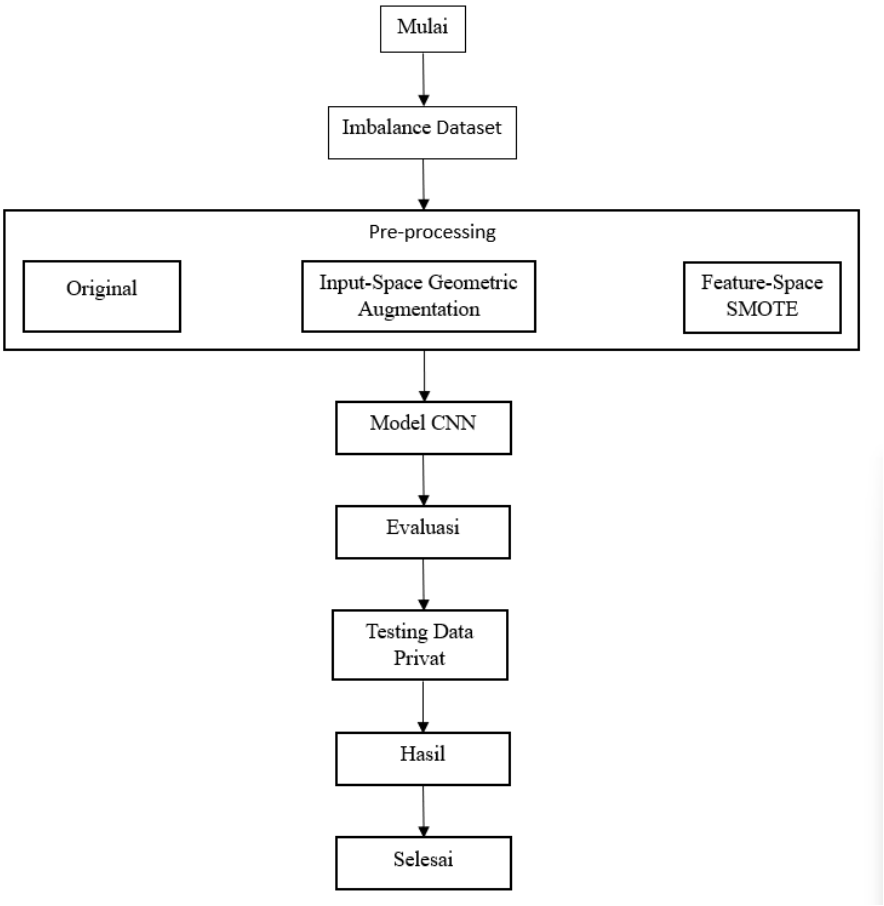
Selain tiga penelitian di atas, ada juga penelitian [14], yang meneliti tentang pengaruh penerapan augmentasi data untuk memperbaiki performa model. Augmentasi dilakukan untuk meningkatkan keragaman data. Pada penelitian tersebut, arsitektur yang digunakan yaitu *ResNet*. Pelatihan model dilakukan dengan dua metode, yaitu *mixup* dan tanpa *mixup*. Hasil penelitan tersebut menyimpulkan bahwa augmentasi data terbukti dapat memperbaiki performa akurasi model, baik pada model *mixup* maupun tanpa *mixup*. Augmentasi data merupakan serangkaian strategi untuk memperbesar dan menyempurnakan dan bentuk suatu gambar dengan tetap mempertahankan labelnya [15]. Augmentasi data digunakan untuk meningkatkan kemampuan pembelajaran model dan menggeneralisasi kinerjanya [16]. Saat ini augmentasi data adalah cara paling efektif untuk mengatasi permasalahan ketersediaan data berkualitas dalam jumlah besar [17].

Berdasarkan beberapa penelitian di atas, permasalahan yang dihadapi yaitu mengatasi *imbalance* dataset dengan berbagai metode. Pada penelitian ini akan dilakukan pembuatan model klasifikasi *CT-Kidney stone* dengan dataset yang *imbalance*. Model akan dibuat dengan 3 teknik berbeda kemudian diuji untuk mengklasifikasi data privat. Tujuan dari menguji dengan data privat yaitu untuk melihat performa dan *overfitting* pada model. Teknik yang pertama yaitu menggunakan data yang *imbalance* untuk membuat model. Teknik yang kedua yaitu melakukan augmentasi data pada dataset minoritas agar data lebih beragam sehingga diharapkan performa model lebih baik. Teknik yang ketiga yaitu melakukan *oversampling* dengan SMOTE untuk menyeimbangkan distribusi data. Penelitian ini bertujuan untuk mencari metode terbaik dalam membuat *modelling* dengan data CT-*Kidney* yang memiliki distribusi data *imbalance*. Tolok ukur dalam menilai hasil *modelling* yaitu dengan tingkat akurasi dan tingkat *overfitting* dari pengujian dengan data privat. Hasil penelitian ini diharapkan dapat menjadi referensi untuk pembuatan model klasifikasi dengan karakteristik data yang mirip dengan CT-*Kidney* dan dengan distribusi yang *imbalance*. Selain itu, penelitian ini dapat dikembangkan lebih jauh lagi untuk menemukan metode yang lebih baik daripada metode yang digunakan pada penelitian ini.

2. METODE PENELITIAN

Penelitian ini dilakukan dengan membuat model dengan algoritma yang sama tetapi dengan *pre-processing* yang berbeda. *Pre-processing* dilakukan dengan 3 metode berbeda, yaitu *original*, augmentasi, dan SMOTE. Dataset yang digunakan adalah dataset yang memiliki *imbalance class*. Perbedaan distribusi kelas antara kelas mayoritas dan kelas minoritas cukup besar sehingga kemungkinan akan mengakibatkan terjadinya *overfitting*. Untuk melihat perbandingan *overfitting* yang terjadi pada tiap model, digunakan data privat untuk menguji model sehingga hasilnya bisa dibandingkan. Berikut alur dari penelitian yang dilakukan.

2.1 Alur Penelitian



Gambar 1. Alur Penelitian

2.2 Dataset

Dataset yang digunakan terdiri dari 2 jenis, dataset publik dan dataset privat. Dataset publik digunakan untuk pelatihan dan evaluasi model. Dataset privat digunakan untuk pengujian model dan melihat tingkat overfitting pada model.

1. Dataset Publik

Dataset publik diambil dari Kaggle (https://www.kaggle.com/datasets/nazmul0087/ct-kidney-dataset-normal-cyst-tumor-and-stone). Data tersebut dikumpulkan dari PACS (Picture Archiving and Communication System) dari berbagai rumah sakit di Kota Dhaka, Bangladesh. Data tersebut juga telah diverifikasi kebenarannya oleh ahli radiologi dan teknolog medis. Sebelum digunakan, data terlebih dahulu dipilah untuk diambil gambar yang bagian spesifik pada ginjal. Data yang digunakan hanyalah kelas normal dan stone. Jumlah data setelah dipilah yaitu data normal 3274 gambar dan data *stone* 848 gambar. Setelah itu, data dibagi menjadi 3 yaitu data *train*, data *val*, dan data *test*. Pembagian dilakukan dengan rasio data *train*=80%, *val*=10%, *test*=10%.

2. Dataset privat

Dataset privat berguna untuk menguji model yang telah dibuat menggunakan data publik. Data tersebut merupakan data asli dari pasien yang berasal dari Indonesia. Data tersebut didapatkan dari tenaga medis dan kebenaran datanya dapat dipertanggungjawabkan, tetapi untuk menjaga privasi, identitas dari pasien dirahasiakan. Dengan pengujian data privat dapat dilihat perbandingan dari 3 model yang telah dibuat. Dataset privat terdiri dari 540 gambar dengan kelas normal dan 446 gambar dengan kelas *stone*.

2.3 Arsitektur CNN

Algoritma yang dipakai untuk pelatihan model yaitu CNN. CNN telah berhasil digunakan untuk mengenali wajah, objek, rambu jalan, serta meningkatkan penglihatan robot dan mobil tanpa pengemudi [18]. CNN bekerja dengan mengambil gambar input dan menentukan tolok ukur tertentu (bobot dan bias yang dapat dipelajari) pada berbagai objek dalam gambar tersebut untuk membedakan satu objek dengan objek lain [19]. Arsitektur CNN terdiri dari *convolutional layer*, *pooling layer*, dan *fully connected layer* [20]. Setiap layer memiliki fungsi yang berbeda-beda. *Convolutional layer* berfungsi untuk mendeteksi pola tertentu. *Pooling layer* berfungsi mereduksi dimensi spasial, mengurangi beban komputasi dan jumlah parameter dalam jaringan. *Fully connected layer* berfungsi menghubungkan setiap neuron dalam satu lapisan ke setiap neuron dalam lapisan berikutnya.

Arsitektur yang digunakan pada penelitian ini yaitu custom, dengan 7 layer konvolusi, 9 layer *batch normalization*, 4 layer *average pooling*, 5 layer *dropout*, dan 2 layer *dense*. *Optimizer* yang digunakan yaitu Adam dengan *learning rate*=0.0001. *Epochs* yang digunakan yaitu 100 tetapi dilengkapi dengan *early stopping*. Berikut gambar dari arsitektur yang digunakan.

Layer	Output Shape	Activation
Conv2D	(None, 128, 128, 32)	Relu
Batch Normalization	(None, 128, 128, 32)	
Conv2D	(None, 128, 128, 32)	Relu
Batch Normalization	(None, 128, 128, 32)	
Average Pooling2D	(None, 64, 64, 32)	
Dropout	(None, 64, 64, 32)	
Conv2D	(None, 64, 64, 64)	Relu
Batch Normalization	(None, 64, 64, 64)	
Conv2D	(None, 64, 64, 64)	Relu
BatchNormalization	(None, 64, 64, 64)	
Average Pooling2D	(None, 32, 32, 64)	
Dropout	(None, 32, 32, 64)	
Conv2D	(None, 32, 32, 32)	Relu
Batch Normalization	(None, 32, 32, 32)	
Average Pooling2D	(None, 16, 16, 32)	
Dropout	(None, 16, 16, 32)	
Conv2D	(None, 16, 16, 64)	Relu
Batch Normalization	(None, 16, 16, 64)	
Conv2D	(None, 16, 16, 64)	Relu
Batch Normalization	(None, 16, 16, 64)	
Average Pooling2D	(None, 8, 8, 64)	
Dropout	(None, 8, 8, 64)	
Flatten	(None, 4096)	
Batch Normalization	(None, 4096)	
Dense	(None, 128)	Relu
Dropout	(None, 128)	
Batch Normalization	(None, 128)	
Dense	(None, 2)	Softmax

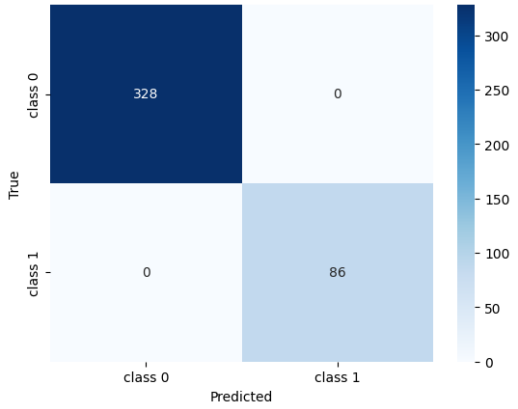
Gambar 2. Arsitektur CNN

3. HASIL DAN PEMBAHASAN

Penelitian ini berfokus pada *pre-processing* sebelum data dipakai untuk melatih model. *Pre-processing* dilakukan dengan 3 cara yang berbeda yaitu *original*, augmentasi, dan *smote oversampling*. Digunakannya 3 cara berbeda tersebut untuk melihat cara mana yang menghasilkan performa paling baik. Sebelum pembuatan model, data terlebih dahulu dibagi menjadi 3 yaitu *train*, *val*, dan *test*. Data *train* berjumlah 3297 gambar, terdiri dari 2619 kelas normal dan 678 kelas *stone*. Data *val* berjumlah 411 gambar dengan 327 merupakan kelas normal dan 84 merupakan kelas *stone*. Data *test* berjumlah 414 gambar yang terdiri dari 328 kelas normal dan 86 kelas *stone*. Penelitian ini menggunakan *epoch*=100 tetapi tetapi pelatihan dapat berhenti sebelum itu dikarenakan menggunakan fungsi *early stopping*. *Early stopping* adalah fungsi untuk menghentikan pelatihan apabila tidak terdapat peningkatan kinerja pada pelatihan tersebut. Pelatihan model menggunakan 3 metode yang berbeda dan memiliki hasil yang berbeda pula, berikut pembahasannya.

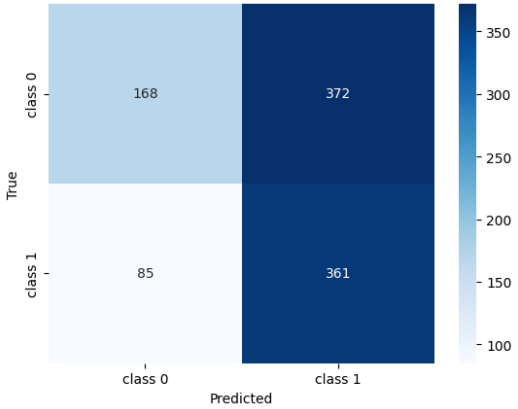
1. Metode Original

Pada metode *original*, data *train* yang masih *imbalance* langsung digunakan untuk pelatihan model. Pelatihan metode original ini berlangsung dengan 14 *epoch*. Pada *testing* model dengan data publik menghasilkan akurasi 100% dengan *loss*=0.0024 dan *f1-score*=1.0. Berikut merupakan *classification report*-nya.



Gambar 3. Classification report data publik metode original

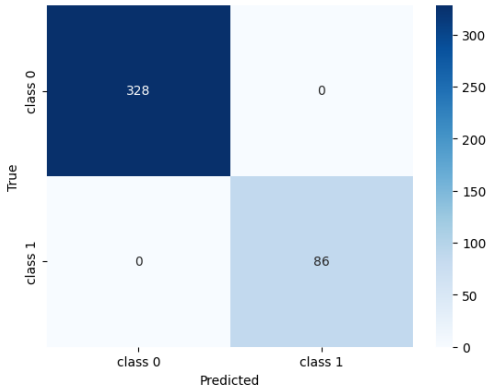
Pada *testing* data privat dengan metode *original* menghasilkan akurasi 53,65% dengan *loss*=2.3637 dan *f1-score*=0.5090. Berikut *classification report*-nya.



Gambar 4. Classification report data privat metode original

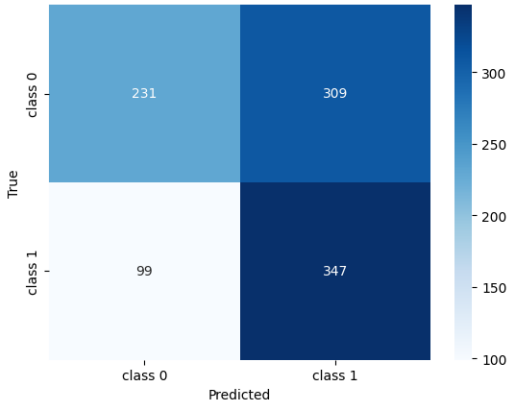
2. Metode Augmentasi Data

Pada metode augmentasi, data minoritas diperbanyak dengan augmentasi agar jumlahnya sama dengan data mayoritas sehingga tercipta dataset yang seimbang dan dapat menghasilkan performa model yang lebih baik. Augmentasi data dilakukan dengan operasi *rotate*, *flip left-right*, *flip top-bottom*. Hasil augmentasi kemudian dipakai untuk pelatihan model. Pelatihan model pada metode augmentasi data berlangsung dengan 6 *epoch*. Hasil dari *testing* model dengan menggunakan data publik didapat akurasi 100% dan *f1-score*=1.0 dengan nilai *loss*=0.0027. Berikut hasil *classification report* data publik dengan metode augmentasi data.



Gambar 5. Classification report data publik metode augmentasi data

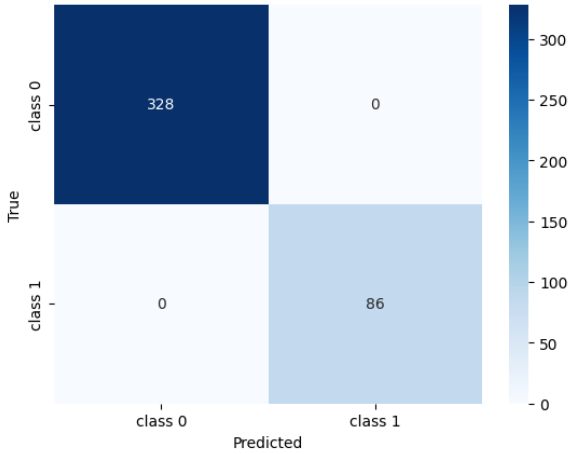
Pada testing metode augmentasi data dengan data privat, didapat hasil akurasi 58,62% dengan loss=1.2333 dan f1-score=0.5756. Berikut classification report-nya.



Gambar 6. Classification report data privat metode augmentasi

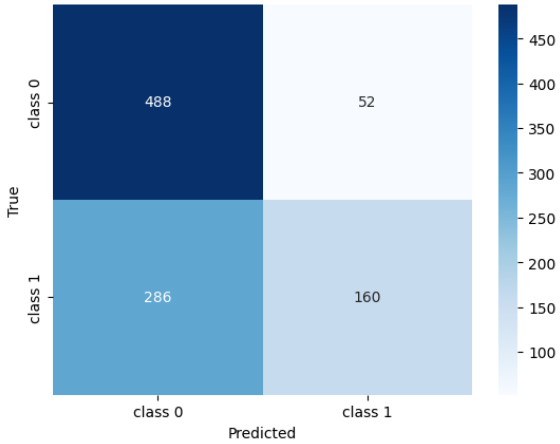
3. Metode SMOTE

Pada metode SMOTE data *train* yang imbalance diatasi dengan cara SMOTE yang disediakan oleh *library imb-learn*. Distribusi data yang awalnya kelas normal=2619 dan *stone*=678 diseimbangkan menjadi sama 2619. Pelatihan model pun dilakukan dengan data hasil SMOTE tersebut, *epoch* yang dijalankan yaitu sebanyak 6. Ketika dilakukan *testing* dengan menggunakan data publik didapat hasil akurasi 100% dan *f1-score*=1.0 dengan loss=0.0006. Ini merupakan hasil akurasi tertinggi dan *loss* terendah pada pengujian data publik dari dua pelatihan sebelumnya. Berikut *classification report* dari metode SMOTE data publik.



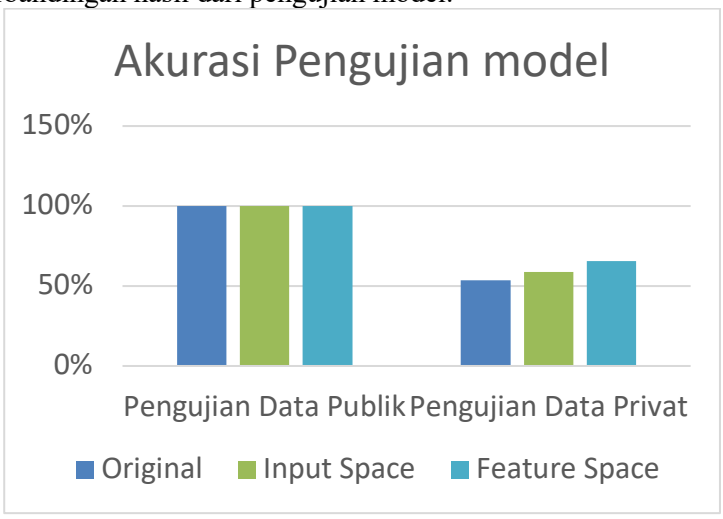
Gambar 7. Classification report data publik metode smote

Pada pengujian metode SMOTE dengan data privat didapatkan hasil akurasi 65,72% dan *f1-score*=0.6267 dengan nilai *loss*=1.3615. Hasil metode SMOTE memiliki akurasi paling tinggi pada pengujian data privat. Berikut *classification report*-nya.



Gambar 8. Classification report data privat metode smote

Berikut sajian perbandingan hasil dari pengujian model.



Gambar 9. Grafik akurasi hasil pengujian

4. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian yang telah dilakukan dengan menggunakan 3 metode yang berbeda, didapat hasil yang terbaik yaitu metode yang ketiga. Pada metode ketiga dataset yang *imbalance* dilakukan SMOTE *oversampling* dengan menggunakan *library imb-learn* sehingga distribusi data bisa seimbang. Metode ketiga terbukti memiliki akurasi yang paling tinggi dibanding dua metode lainnya dengan selisih akurasi yang cukup signifikan, tingkat akurasi metode ketiga yaitu 65,72% berbanding dengan metode kedua 58,62% dan metode pertama 53,65%. Dengan tingginya akurasi pada metode ketiga menunjukkan bahwa metode ketiga merupakan model yang paling baik dalam mengatasi adanya *overfitting* yang terjadi pada pembuatan model dengan data *imbalance*.

DAFTAR PUSTAKA

[1] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Mark.*, vol. 31, no. 3, pp. 685–695, 2021, doi: 10.1007/s12525-021-00475-2.

[2] J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, "A review of convolutional neural network applied to fruit image processing," *Appl. Sci.*, vol. 10, no. 10, p. 3443, May 2020, doi: 10.3390/app10103443.

[3] E. MUGABO and D. W. M. (PhD), "Develop an Extended Model of CNN Algorithm in Deep Learning for Bone Tumor Detection and its Application," *Int. J. Innov. Sci. Res. Technol.*, vol. 8, no. 10, 2023, doi: https://doi.org/10.5281/zenodo.10040584.

[4] O. Joseph and W. O. Apena, "Development of Segmentation and Classification Algorithms for Computed Tomography Images of Human Kidney Stone," *J. Electron. Res. Appl.*, vol. 5, no. 5, pp. 1–10, 2021, doi: 10.26689/jera.v5i5.1196.

[5] A. Indrawati, "Penerapan Teknik Kombinasi Oversampling Dan Undersampling Hybrid Oversampling and Undersampling Techniques To Handling Imbalanced Dataset," *JIKO(Jurnal Inform. dan Komputer)*, vol. 4, no. 1, pp. 38–43, 2021, doi: 10.33387/jiko.

[6] X. Ying, "An Overview of Overfitting and its Solutions," *J. Phys. Conf. Ser.*, vol. 1168, no. 2, 2019, doi: 10.1088/1742-6596/1168/2/022022.

[7] Y. Yan *et al.*, "Oversampling for imbalanced data via optimal transport," *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019*, vol. 33, no. 1, pp. 5605–5612, 2019, doi: 10.1609/aaai.v33i01.33015605.

[8] T. Wongvorachan, S. He, and O. Bulut, "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining," *Inf.*, vol. 14, no. 1, 2023, doi: 10.3390/info14010054.

[9] S. Bej, N. Davtyan, M. Wolfien, M. Nassar, and O. Wolkenhauer, "LoRAS: an oversampling approach for imbalanced datasets," *Mach. Learn.*, vol. 110, no. 2, pp. 279–301, 2021, doi: 10.1007/s10994-020-05913-4.

[10] C. Supriyanto, A. Salam, J. Zeniarja, and A. Wijaya, "Two-Stage Input-Space Image Augmentation and Interpretable Technique for Accurate and Explainable Skin Cancer Diagnosis," *Computation*, vol. 11, no. 12, p. 246, Dec. 2023, doi: 10.3390/computation11120246.

[11] A. N. Kasanah, M. Muladi, and U. Pujianto, "Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 196–201, 2019, doi: 10.29207/resti.v3i2.945.

[12] G. Gumelar, Q. Ain, R. Marsuciati, S. Agustanti Bambang, A. Sunyoto, and M. Syukri Mustafa, "Kombinasi Algoritma Sampling dengan Algoritma Klasifikasi untuk Meningkatkan Performa Klasifikasi Dataset Imbalance," *SISFOTEK Sist. Inf. dan Teknol.*, vol. 5, no. 1, pp. 250–255, 2021.

[13] A. Nugroho and E. Rilvani, "Penerapan Metode Oversampling SMOTE Pada Algoritma

Random Forest Untuk Prediksi Kebangkrutan Perusahaan," *Techno.Com*, vol. 22, no. 1, pp. 207–214, 2023, doi: 10.33633/tc.v22i1.7527.

[14] J. Sanjaya and M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.

[15] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 91–99, 2022, doi: 10.1016/j.gltp.2022.04.020.

[16] D. Alzu'Bi *et al.*, "Kidney Tumor Detection and Classification Based on Deep Learning Approaches: A New Dataset in CT Scans," *J. Healthc. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/3861161.

[17] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, no. November, p. 100258, 2022, doi: 10.1016/j.array.2022.100258.

[18] R. Setya Nugraha and A. Hermawan, "Optimasi Akurasi Metode Convolutional Neural Network Untuk Klasifikasi Kualitas Buah Apel Hijau," *J. Mnemon.*, vol. 6, no. 2, pp. 149–156, 2023, doi: 10.36040/mnemonic.v6i2.6730.

[19] D. Bhatt *et al.*, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electron.*, vol. 10, no. 20, pp. 1–28, 2021, doi: 10.3390/electronics10202470.

[20] M. Resa Arif Yudianto, P. Sukmasetya, R. Abul Hasani, and D. Sasongko, "Pengaruh Data Preprocessing terhadap Imbalanced Dataset pada Klasifikasi Citra Sampah menggunakan Algoritma Convolutional Neural Network," *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, pp. 1367–1375, 2022, doi: 10.47065/bits.v4i3.2575.