IJEIS (Indonesian Journal of Electronics and Instrumenations Systems)
Vol.13, No.1, April 2023, pp. 13~22
ISSN (print): 2088-3714, ISSN (online): 2460-7681
DOI: 10.22146/ijeis.79536

Klasifikasi Suara Untuk Memonitori Hutan Berbasis *Convolutional Neural Network*

Rizqi Fathin Fadhillah*[1], Raden Sumiharto[2]
[1]Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta, Indonesia
[2]Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia
e-mail: *[1]**rizqifathin@mail.ugm.ac.id**, [2]r_sumiharto@ugm.ac.id

Abstrak

Hutan memiliki peran yang penting bagi kehidupan di muka bumi. Kebutuhan untuk memonitori hutan terhadap adanya aktivitas illegal dan jenis-jenis satwa yang ada disana perlu dilakukan untuk menjaga kondisi hutan agar tetap baik. Akan tetapi, kondisi hutan yang luas dan keterbatasan sumber daya membuat pemantauan kondisi hutan yang dilakukan secara langsung oleh petugas (manusia) menjadi terbatas. Dalam hal ini, suara dengan pemrosesan sinyal digital dapat digunakan sebagai salah satu sarana untuk memonitori hutan. Pada penelitian ini, diimplementasikan sistem untuk mengklasifikasikan suara pada Raspberry Pi 3B+. Suara yang diklasifikasikan meliputi suara gergaji mesin, suara tembakan, dan suara dari 8 jenis burung. Tahapan dimulai dari proses akuisisi data yang dilakukan dengan mengumpulkan data dari internet dan menerapkan beberapa augmentasi data. Sinyal audio kemudian direpresentasikan dalam bentuk mel-spektogram. Tahapan berikutnya adalah ekstraksi fitur yang dilakukan menggunakan pretrained VGG-16 dan MobileNetV3. Tahapan terakhir adalah klasifikasi menggunakan metode Random Forest, SVM, KNN, dan MLP. Berdasarkan hasil penelitian ini, didapatkan bahwa model MobileNetV3-Small + MLP dengan data latih gabungan dari augmentasi time stretch dan time shift memberikan performa yang paling bagus untuk diimplementasikan pada sistem ini, dengan durasi inference 0.8 detk; akurasi sebesar 93.96%; dan presisi sebesar 94.1%.

Kata kunci*—Klasifikasi suara, Mel-spectogram, CNN, MLP*

Abstract

Forest has an important role on earth. The need to monitor forest from illegal activities and the types of animals in there is needed to keep the forest in good condition. However, the condition of the vast forest and limited resource make direct forest monitoring by officer (human) is limited. In this case, sound with digital signal processing can be used as a tool for forest monitoring. In this study, a system was implemented to classify sound on the Raspberry Pi 3B+ using mel-spectrogram. Sounds that classified are the sound of chainsaw, gunshot, and the sound of 8 species of bird. This study also compared pretrained VGG-16 and MobileNetV3 as feature extractor, and several classification methods, namely Random Forest, SVM, KNN, and MLP. To vary and increase the number of training data, we used several types of data augmentation, namely add noise, time stretch, time shift, and pitch shift. Based on the result of this study, it was found that the MobileNetV3-Small + MLP model with combined training data from time stretch and time shift augmentation provide the best performance to be implemented in this system, with an inference duration of 0.8 seconds; 93.96% accuracy; and 94.1% precision.

Keywords— *Sound classification, Mel-spectogram, CNN, MLP*

Received November 29th,2022; Revised April 30th, 2023; Accepted April 30th, 2023

1. PENDAHULUAN

Hutan memiliki peran yang penting bagi kehidupan di muka bumi. Hutan berfungsi sebagai cadangan sumber energi di bumi dan memainkan peran penting sebagai pengendali cuaca dan pengatur berbagai siklus air [1]. Akan tetapi, seiring berjalannya waktu, hutan terus mengalami penyusutan. Laju deforestasi di Indonesi mengalami peningkatan dan penurunan disetiap periodenya [2]. Pada periode tahun 2018-2019 terjadi deforestasi sebesar 0.46 juta ha. Hal itu terjadi karena dinamisnya perubahan dan penutupan lahan akibat aktivitas manusia seperti illegal logging.

Hutan juga berperan sebagai salah satu habitat dari berbagai jenis flora dan fauna yang ada di bumi. Ancaman terhadap hutan akan berpotensi pada kelangkaan atau bahkan kepunahan dari flora dan fauna tersebut. Selain karena berkurangnya hutan, ancaman kepunahan satwa juga terjadi karena adanya perburuan liar untuk perdagangan. Perdagangan satwa liar menjadi ancaman yang serius bagi kelangsungan hidup satwa di alam karena lebih dari 95% satwa yang diperdagangkan berasal dari tangkapan alam dan sisanya hasil penangkaran [3].

Beberapa upaya untuk mencegah aktivitas ilegal di hutan telah dilakukan seperti patroli oleh petugas setempat. Akan tetapi upaya tesebut mengalami beberapa kendala, diantaranya adalah sumber daya manusia yang kurang, sarana dan prasarana yang kurang mendukung, dan keterbatasan dana [4]. Patroli yang dilakukan oleh manusia juga terbatas pada waktu tertentu, sehingga pada waktu yang lain masih berpotensi adanya tindakan ilegal. Salah satu alternatif yang dapat dilakukan adalah dengan menerapkan system untuk memonitori hutan berdasarkan suara.

Pada penelitian [5], digunakan beberapa pretrained CNN untuk mengklasifikasikan suara burung, kucing, dan suara lingkungan. Hasil dari penelitian tersebut menunjukan VGG-16 dan VGG-19 memberikan performa *stand-alone* CNN terbaik. Akan tetapi, pada penelitian [6] yang mengimplementasikan klasifikasi citra menggunakan transfer learning VGG-16 pada Raspberry Pi 3 didapatkan durasi komputasi yang cukup lama, sekitar 1.5 - 2 detik. Dalam hal ini, salah satu *pretrained* CNN yang dapat digunakan adalah MobileNetV3 [7] karena memberikan beban komputasi yang cukup rendah dan biasa digunakan pada *mobile device*.

Dari pemaparan diatas, proses identifikasi jenis suara dibutuhkan untuk membantu memonitori kondisi hutan. Raspberry Pi 3B+ akan digunakan sebagai *device* untuk mengkomputasi sinyal suara karena kemampuannya yang cukup mumpuni untuk melakukan tugas ini. Augmentasi data audio yang meliputi *add noise, time stretch, time shift,* dan *pitch shift* akan diterapkan pada penelitian ini untuk memperbanyak dan memvariasikan data pada proses pelatihan model. Representasi *time-frequency* yang akan digunakan adalah mel-spektogram. Selain itu juga akan digunakan beberapa *pre-trained* arsitektur CNN, yaitu VGG-16 dan MobileNetV3 sebagai *feature extractor* dan untuk *classifier*-nya digunakan beberapa metode klasifikasi, yaitu SVM, KNN, Random Forest, dan MLP.

2. METODE PENELITIAN

Pada penelitian ini, akan dibuat sistem yang mampu mengklasifikasikan sinyal suara kedalam 10 kelas, yaitu suara gergaji mesin, tembakan senjata, dan suara dari 8 jenis burung pada Raspberry Pi 3B+. Tahapan dimulai dari proses akuisisi data yang dilakukan dengan mengumpulkan data dari internet dan menerapkan beberapa augmentasi data. Sinyal audio kemudian direpresentasikan dalam bentuk mel-spektogram. Tahapan berikutnya adalah ekstraksi fitur yang dilakukan menggunakan *pretrained* VGG-16 dan MobileNetV3. Tahapan terakhir adalah klasifikasi menggunakan metode Random Forest, SVM, KNN, dan MLP. Diagram blok sistem ditunjukan pada Gambar 1.

Gambar 1 Diagram blok sistem

2.1. Akuisisi Data

Dataset yang digunakan pada penelitian ini berasal dari dataset ESC-50 [8] untuk suara gergaji mesin, UrbanSound8K [9] untuk suara tembakan senjata, dan BirdClef-2021 [10] untuk 8 jenis suara burung (Wild Turkey, Black-bellied Plover, American Coot, Great Crested Flycatcher, Townsend's Solitaire, Ruddy Turnstone, Common Chlorospingus, dan Black-andwhite Warbler). Data audio yang telah dikumpulkan kemudian akan disamakan formatnya, meliputi format file .wav, *sampling rate* 32 kHz, *mono-channel*, *bit depth* 16 bit, dan durasi sinyal 1,3, dan 5 detik. Sinyal audio yang memiliki durasi lebih dari 1 detik dan kurang dari n detik akan ditambahkan sinyal dengan amplitudo 0 (*zero padding*), sehingga durasinya akan menjadi n detik. Sinyal dengan durasi kurang dari 1 detik tidak akan digunakan, tetapi apabila dalam satu *file* hanya terdapat sinyal dengan durasi kurang dari 1 detik, maka sinyal tersebut tetap digunakan dengan menambahkan sinyal dengan amplitudo 0 (*zero padding*), sehingga durasinya menjadi n detik. Hal itu dilakukan untuk menghindari tidak adanya suara yang dibutuhkan pada data tersebut. Nilai n menunjukan durasi pemotongan sinyal. Pada proses ini juga dilakukan augmentasi data untuk data latih, yaitu *add noise* dengan faktor [0.001, 0.015], *time stretch* dengan faktor [0.8, 1.25], *time shift* dengan faktor [-3200, 3200], dan *pitch shift* dengan faktor [-4, 4]. Kombinasi dari beberapa jenis augmentasi data akan dilakukan berdasarkan pengaruh dari setiap jenis augmentasi data terhadap performa model klasifikasi. Penggunaan augmentasi data dimaksudkan untuk memvariasi dan memperbanyak data latih.

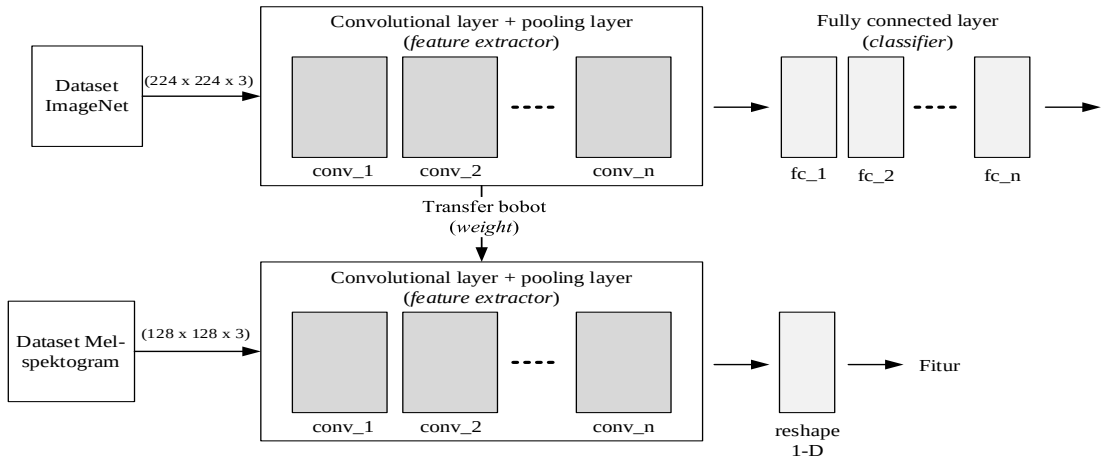
2.2 Pra-pengolahan

Mel-spektogram merupakan salah satu representasi *time-frequency* yang biasa digunakan dalam kasus pengenalan suara. Proses mel-spektogram [11] dimulai dari sampling sinyal kedalam jendela yang terpisah dengan menerapkan teknik *framing* dan *windowing*. Fungsi *window* yang digunakan pada penelitian ini adalah *hanning window* dengan panjang sama seperti panjang *frame*, yaitu 1024. *Hop size* yang digunakan sebesar 251, 755, dan 1259 masing-masing untuk sinyal dengan durasi 1,3, dan 5 detik. Fast Fourier Transform (FFT) dari setiap *frame* kemudian dihitung untuk mentransformasikan sinyal dalam domain waktu ke domain frekuensi. Terakhir, mel-spektogram dihasilkan untuk setiap *window* dengan menguraikan besarnya sinyal menjadi komponen-komponennya, sesuai dengan frekuensi dalam skala mel. Konversi frekuensi kedalam skala mel dilakukan dengan menggunakan 128 *mel-filter bank*. Sinyal hasil akuisisi data kemudian akan dikonversi kedalam bentuk mel-spektogram. Proses konversi dilakukan menggunakan pustaka Librosa dengan rentang frekuensi yang digunakan 100 Hz – 15 kHz. Data mel-spektogram kemudian akan diestimasi nilai *snr*-nya mengikuti [12] dengan *threshold* 0.0001. Data yang tidak memenuhi nilai *threshold* tidak akan dipakai, sedangkan data yang memenuhi akan disimpan dalam bentuk citra *grayscale*.

2.3 Ekstraksi Fitur

CNN atau *Convolutional Neural Network* memiliki 3 layer utama, yaitu *convolutional layer, pooling layer,* dan *fully connected layer* [13]. Pada umumnya, CNN membutuhkan data dengan jumlah besar untuk dapat melatih model dengan akurat dan tidak biasa untuk melakukan pelatihan pada dataset yang berjumlah puluhan atau bahkan ratusan ribu citra [14], sehingga salah satu solusi yang dapat digunakan adalah *transfer learning* dari model yang telah dilatih sebelumnya. Pada penelitian ini digunakan 3 *pretrained* CNN yaitu VGG-16 [15], MobileNetV3-

Small [7] dan MobileNetV3-Large [7] yang telah dilatih pada dataset ImageNet. Ilustrasi proses *transfer learning* ditunjukan pada Gambar 2.



Gambar 2 Diagram proses *transfer leaning*

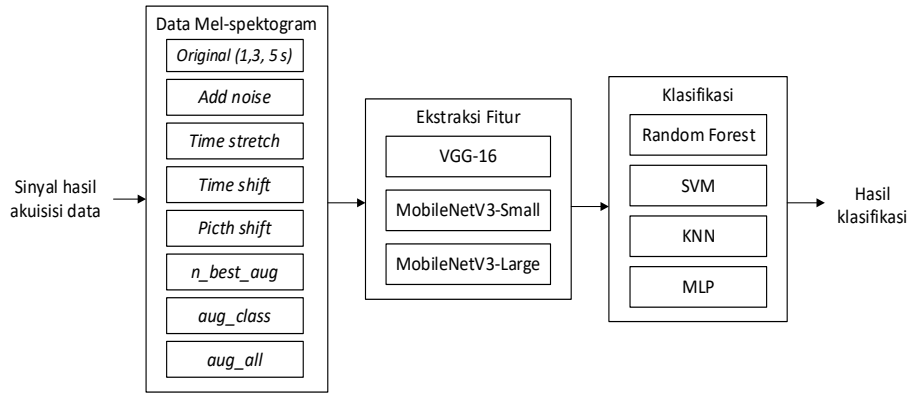
Sebelum diekstraksi fitur, citra *grayscale* akan dirubah menjadi 3 *channel* dengan menyalin nilai citra *grayscale* pada *channel* yang lain, sehingga ukuran data menjadi (128,128,3). *Layer* dari model *pretrained* yang akan digunakan adalah *convolutional* dan *pooling layer* yang merupakan *feature extractor layer*. Keluaran dari model *pretrained* merupakan *array* multi-dimensi, sehingga perlu ditransformasi menjadi *array* 1-D untuk setiap data.

2.4 Klasifikasi

Terdapat 4 metode klasifikasi yang akan digunakan, yaitu Random Forest, SVM, KNN, dan MLP. Pustaka Scikit-learn akan digunakan untuk mengimplementasikan metode klasifikasi tersebut. Parameter yang akan digunakan pada keempat model akan ditala menggunakan fungsi GridSearchCV dengan MobileNetV3-Small sebagai *feature extractor*-nya. Parameter tersebut antara lain: jumlah pohon pada metode Random Forest; nilai C pada metode SVM; jumlah tetangga pada metode KNN; *hidden layer* dan *learning rate* pada metode MLP. Pada metode MLP, *input layer* akan berupa variabel dengan jumlah node menyesuaikan ukuran data hasil ekstraksi fitur dari VGG16, MobileNetv3-Small, dan MobileNetv3-Large yang masing-masing berjumlah 8192, 9216, dan 15360. *Output layer* pada arsitektur jaringan ini memiliki 10 *node* yang menunjukan jumlah kelas yang akan diklasifikasikan. Fungsi aktivasi yang digunakan pada penelitian ini adalah ReLu (*Rectified Linear Unit*) yang berguna untuk merubah nilai *weighted sum* pada neuron agar menjadi *non-linear*. *Optimizer* yang akan digunakan adalah Adam dengan *learning rate* konstan. Proses pelatihan akan berjalan sebanyak 30 epoch. Parameter yang tidak didefinisikan pada tahap ini akan menggunakan parameter *default* dari fungsi yang digunakan.

2.5 Pengujian

Secara keseluruhan variasi pengujian yang dilakukan pada penelitian ini meliputi kombinasi augmentasi pada data latih, penggunaan *pretrained* VGG-16 dan MobileNetv3, dan beberapa metode klasifikasi, yaitu Random Forest, SVM, KNN, dan MLP. Diagram variasi pengujian ditunjukan pada Gambar 3. Setiap data augmentasi yang dimaksudkan disini merupakan data *original* ditambah dengan data augmentasi tersebut. Augmentasi "*n_best_aug*" menunjukan gabungan dari beberapa jenis augmentasi yang memberikan pengaruh positif pada keseluruhan model klasifikasi, "aug_class" menunjukan penerapan augmentasi berdasarkan pengaruh setiap jenis augmentasi data terhadap performa pada masing-masing kelas, sedangkan "aug_all" menujukan gabungan dari seluruh jenis augmentasi data.



Gambar 3 Diagram variasi pengujian sistem

Perbandingan performa dari masing-masing kombinasi model akan dilakukan dengan menggunakan data hasil pengujian *cross validation*. Metode *cross validation* yang digunakan pada penelitian ini adalaha *stratified k-fold* dengan 5 *fold.* Model terbaik kemudian akan diuji menggunakan data uji untuk mengetahui performa model pada data diluar data latih. Ilustrasi pembagian data ditunjukan pada Gambar 3. Hasil pengujian model akan disajikan dalam bentuk *confusion matrix*. Dari *confusion matrix* tersebut kemudian bisa didapatkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), sehingga dapat dihitung nilai akurasi, presisi, *recall*, dan *f1-score* dengan persamaan (1), (2), (3), dan (4). Nilai rata-rata untuk yang digunakan pada setiap *metric* merupakan nilai *weighted average* yang ditunjukan pada persamaan (5). Model juga akan diuji pada Raspberry Pi 3B+ untuk mengetahui durasi *inference* dan daya yang digunakan. Pengukuran daya dilakukan menggunakan sensor INA219 selama 5 menit.

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

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

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1score = \frac{TP}{TP + 0.5\,(FP + FN)} \quad (4)$$

$$weighted\ average = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \quad (5)$$

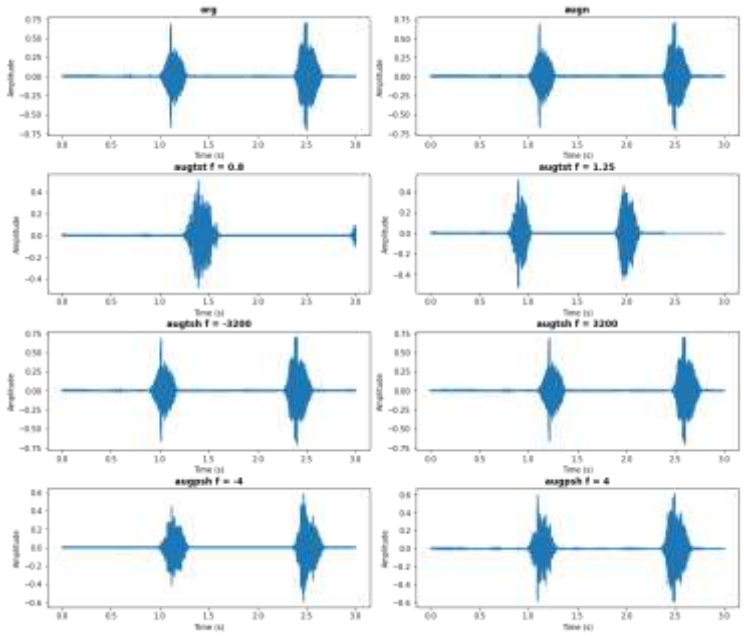
Keterangan:

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

n = jumlah kelas yang akan dihitung nilai rata-ratanya
w= bobot pada setiap kelas
x = nilai pada setiap kelas yang akan dihitung rata-ratanya

3. HASIL DAN PEMBAHASAN

Akuisisi data pada penelitian ini dilakukan dengan mengumpulkan data dari dataset *public* yang tersedia secara gratis di internet. Data audio yang dikumpulkan berasal dari dataset ESC-50, UrbanSound8K, dan BirdClef-2021. Seluruh data yang berhasil dikumpulkan memiliki total durasi sebesar 10,274 detik. Data tersebut kemudian akan distandarkan dan dilakukan augmentasi data *add noise, time stretch, time shift,* dan *pitch shift*. Sinyal hasil dari proses ini ditunjukan pada Gambar 4.



Gambar 4 Sinyal sebelum dan sesudah dilakukan augmentasi data

Sinyal hasil akuisisi data kemudian akan dilakukan pra-pengolahan dengan mengkonversi sinyal kedalam bentuk mel-spektogram dan estimasi *snr*. Data mel-spektogram dengan *snr* < 0.0001 tidak akan digunakan, dan yang lain akan disimpan dalam bentuk citra *grayscale* berukuran 128 x 128. Data latih *original* yang dihasilkan untuk sinyal berdurasi 1,3, dan 5 masing-masing sebanyak 7627, 2818, dan 1874 data.

Data mel-spektogram kemudian diekstrak fiturnya menggunakan *pretrained* VGG-16, MobileNetV3-Small, dan MobileNetV3-Large. Input dari *pretrained* tersebut harus berupa data citra 3 *channel* (RGB), sehingga citra mel-spektogram yang berupa *grayscale* akan dikonversi menjadi RGB dengan menyalin nilai citra tersebut untuk dijadikan sebagai nilai pada *channel* yang lain. *Pretrained* model akan disimpan dalam format .h5 untuk nantinya diimplementasikan pada raspberry Pi 3B+. Keterangan hasil dari ekstraksi fitur ini ditunjukan pada Tabel 1. Kolom durasi pada tabel tersebut menunjukan durasi ekstraksi fitur pada keseluruhan data latih.

Tabel 1 Hasil ekstraksi fitur menggunakan *pretrained* CNN

Feature Extractor	Input	*Output*	Total Parameter	Durasi (detik)	Ukuran (MB)
VGG-16	(n,128,128,3)	(n,4,4,512)	14,714,688	239.379	56.1
MobileNetV3-Small		(n,4,4,576)	939,120	9.029	4.21
MobileNetV3-Large		(n,4,4,960)	2,996,352	18.47	12.1

Nilai n pada kolom input dan *output* menunjukan banyaknya data yang digunakan. Keluaran dari *pretrained model* yang berupa *array multi-dimensi*, kemudian akan di *reshape* menjadi array 1-D untuk setiap data (n), sehingga hasil akhirnya berukuran (n,8192), (n,9216), dan (n,15360) masing-masing untuk *pretrained* VGG-16, MobileNetV3-Small, dan MobileNetV3-Large. VGG-16 memiliki total parameter paling besar yang berakibat pada durasi komputasi yang paling lama dan ukuran model yang paling besar. *Pretrained* yang dengan durasi yang paling cepat adalah MobileNetV3-Small.

Data hasil ekstraksi fitur kemudian akan dijadikan sebagai masukan pada proses klasifikasi. Terdapat 4 metode klasifikasi yang digunakan, yaitu Random Forest (RF), SVM, KNN, dan MLP. Parameter pada keempat metode tersebut didapatkan dari tuning GridSearchCV, yaitu *n_estimators* = 1000 untuk metode Random Forest; C = 10 untuk metode SVM; *n_neighbors* = 1 untuk metode KNN; *hidden layer* = (512,256) dan *learning rate* 0.001 untuk metode MLP. Hasil performa model dengan variasi durasi ditunjukan pada Tabel 2. *Feature extractor* yang digunakan pada pengujian ini adalah MobileNetV3-Small dengan data latih *original*.

Tabel 2 Performa model terhadap variasi durasi sinyal – *cross validation*

Durasi (detik)	*Classifier*	Akurasi	Presisi	*F1-score*
1	RF	82.1	82.97	82.05
	SVM	92.71	92.73	92.7
	KNN	88.49	88.79	88.54
	MLP	92.08	92.1	92.06
3	RF	82.30	82.83	82.21
	SVM	92.11	92.29	92.10
	KNN	87.46	87.89	87.42
	MLP	91.08	91.21	91.07
5	RF	82.07	82.67	81.83
	SVM	90.23	90.42	90.2
	KNN	84	84.61	83.92
	MLP	89.01	89.21	88.93

Hasil tersebut menunjukan sinyal dengan durasi 1 detik cenderung memberikan hasil yang paling bagus, sedangakan sinyal dengan durasi 5 detik memberikan hasil yang paling rendah. Akan tetapi, pada penelitian ini dipilih durasi 3 detik untuk menghindari durasi sinyal sampling yang lebih cepat dari pada durasi komputasi pada Raspberry Pi 3B+ yang kemudian dapat menyebabkan akumulasi *delay* antara waktu sampling dengan waktu nyata. Selain itu, durasi 3 detik juga memberikan hasil yang hampir sama dengan durasi 1 detik. Jumlah data latih dan data uji yang akan digunakan untuk pengujian selanjutnya dengan durasi sinyal 3 detik ditunjukan pada Tabel 3.

Tabel 3 Jumlah data latih dan data uji yang digunakan

Label	Data Latih					Data Uji
	Original	Augmentasi *add noise*	Augmentasi *time stretch*	Augmentasi *time shift*	Augmentasi *pitch shift*	
chainsaw	64	128	192	192	192	16
gun_shot	312	624	936	936	936	78
wiltur	325	650	975	975	975	81
bkbplo	284	568	852	852	852	71
y00475	285	570	855	855	855	71
grcfly	286	572	858	858	858	71
towsol	316	632	948	948	948	79
rudtur	330	660	990	990	990	82
cobtan1	318	636	954	954	954	80
bawwar	298	596	894	894	894	74
Total	2818	5636	8454	8454	8454	703

Pengujian selanjutnya dilakukan untuk mengetahui performa model terhadap variasi *feature extractor*. Pengujian ini dilakukan dengan menggunakan data latih *original.* Hasil pengujian ditunjukan pada Tabel 4. Kolom durasi pada tabel tersebut menunjukan total durasi *inference* (akuisisi data, pra-pengolahan, ekstraksi fitur, dan klasifikasi) pada Raspberry Pi 3B+.

Tabel 4 Performa model pada data latih *original – cross validation*

Feature Extractor	*Classifier*	Akurasi (%)	Presisi (%)	*F1-Score* (%)	Ukuran (MB)	Durasi (detik)	Daya (mW)
VGG-16	RF	86.17	86.50	86.09	135	3.72	3102
	SVM	91.93	92.09	91.93	130	2.93	3222
	KNN	87.92	88.52	87.99	88	3.28	3305
	MLP	89.80	89.95	89.80	49.5	2.56	3308
MobileNet V3-Small	RF	82.30	82.83	82.21	121	1.97	2654
	SVM	92.11	92.29	92.10	154	1.12	2792
	KNN	87.46	87.89	87.42	99	1.67	3016
	MLP	91.08	91.21	91.07	55.5	0.8	2879
MobileNet V3-Large	RF	83.90	84.30	83.82	124	2.12	2690
	SVM	91.86	92.05	91.86	254	1.57	2782
	KNN	87.74	87.96	87.73	165	2.37	3046
	MLP	91.61	91.71	91.60	91.5	0.96	2929

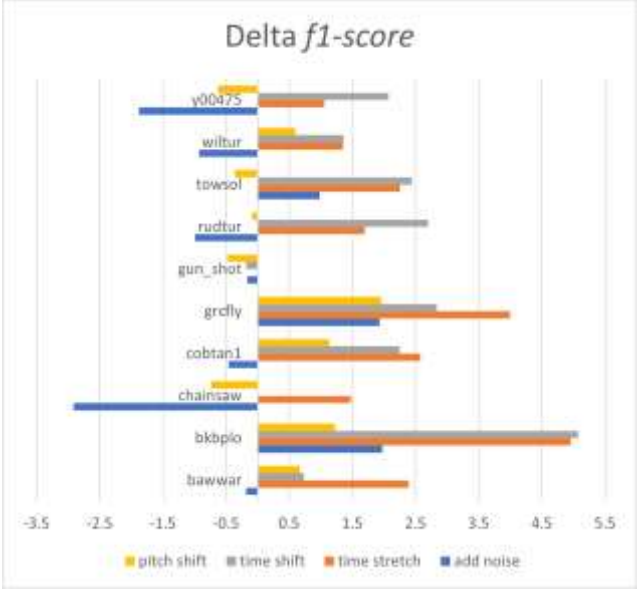
Ditinjau dari performa klasifikasinya (akurasi, presisi, *f1-score*), pengaruh *pretrained* model VGG-16 dan MobileNetV3 tidak terlihat secara konsisten mengungguli *pretrained* model satu dengan yang lain untuk setiap metode klasifikasi yang dikombinasikan, sehingga dapat disimpulkan bahwa ketiga arsitektur tersebut memberikan performa klasifikasi yang hampir sama. Akan tetapi, pada durasi *inference* dan daya yang digunakan, MobileNetV3-Small memberikan hasil yang paling baik, dengan durasi 0.8 detik dan daya 2879 mW pada metode MLP, sehingga secara keseluruhan MobileNetV3-Small lebih unggul dari pada *pretrained* lain yang diuji. Dari 4 metode klasifikasi yang diuji, perbedaan performa klasifikasi terlihat secara konsisten pada *feature extractor* yang sama, dimana metode SVM memiliki performa klasifikasi yang paling baik diikuti dengan metode MLP. Hasil tersebut menunjukan bahwa pemilihan metode klasifikasi memiliki pengaruh yang cukup signifikan terhadap performa klasifikasi model.

Pengujian selanjutnya dilakukan untuk mengetahui pengaruh dari setiap jenis augmentasi data terhadap performa model. Pada pengujian ini digunakan *feature extractor* MobileNetV3-Small dengan metode klasifikasi SVM dan MLP yang memberikan performa terbaik pada pengujian sebelumnya. Hasil pengujian ini ditunjukan pada Tabel 5.

Tabel 5 Performa model pada data latih setiap jenis augmentasi – *cross validation*

Dataset	*Classifier*	Akurasi (%)	Presisi (%)	*F1-Score* (%)	Ukuran (MB)	Durasi (detik)
add noise	SVM	91.86	91.93	91.84	279	1.52
	MLP	91.04	91.15	91.01	55.5	0.81
time stretch	SVM	93.89	93.97	93.89	397	1.85
	MLP	93.28	93.36	93.27	55.5	0.82
time shift	SVM	94.03	94.13	94.02	397	1.79
	MLP	93.14	93.27	93.14	55.5	0.83
pitch shift	SVM	92.08	92.19	92.05	397	1.84
	MLP	91.51	91.66	91.48	55.5	0.81

Hasil pengujian menunjukan penggunaan augmentasi data dapat meningkatkan performa klasifikasi model, sebagaimana ditunjukan pada augmentasi *time stretch* dan *time shift*. Akan tetapi, terdapat beberapa jenis augmentasi data tidak memberikan peningkatan yang signifikan dan pada beberapa model malah memberikan penurunan perfoma klasifikasi seperti yang ditunjukan pada augmentasi *add noise* dan *pitch shift*. Hasil tersebut juga menunjukan bahwa metode SVM memberikan peningkatan ukuran model dan durasi *inference* seiring bertambahnya jumlah dataset, sedangkan pada metode MLP tetap sama, sehingga dalam kasus ini metode MLP lebih diunggulkan dari pada metode SVM. Pengaruh setiap jenis augmentasi data pada model MobileNetV3-Small + MLP ditunjukan pada Gambar 5. Dari hasil tersebut terlihat bahwa setiap kelas mendapatkan pengaruh yang berbeda-beda untuk jenis augmentasi yang sama.



Gambar 5 Perbedaan performa data *original* dengan data augmentasi pada setiap kelas

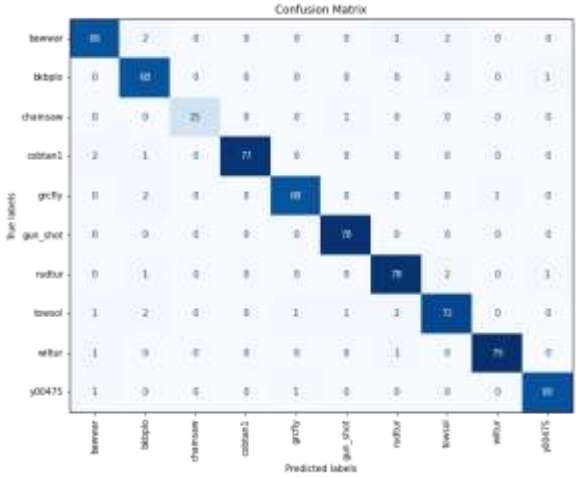
Pengujian selanjutnya dilakukan dengan menggunakan data latih gabungan dari beberapa jenis augmentasi data. Hasil dari pengujian ini ditunjukan pada Tabel 6. Baris "augtst_tsh" menunjukan performa dari data latih gabungan augmentasi *time stretch* dan *time shift*, "aug_class" menunjukan performa dari penerapan augmentasi data pada kelas-kelas tertentu sesuai hasil pada Gambar 5, sedangkan "aug_all" merupakan performa dari data latih gabungan dari seluruh jenis augmentasi.

Tabel 6 Performa model gabungan dari beberapa data augmentasi

Dataset	Akurasi	Presisi	*F1-score*
augtst_tsh	93.96	94.05	93.95
aug_class	93.78	93.92	93.78
aug_all	93.78	93.90	93.77

Keseluruhan dataset yang diuji memberikan hasil yang mirip dan mengalami peningkatan dari penggunaan augmentasi data tunggal. Selain itu penggunaan beberapa jenis augmentasi data tertentu yang memberikan peningkatan yang siginifikan lebih efisien untuk digunakan dari pada dengan keseluruhan jenis augmentasi data, karena data latih yang digunakan akan lebih sedikit dengan performa yang setara. Dari pengujian ini didapatkan hasil terbaik pada model MobileNetV3-Small + MLP dengan data latih gabungan dari augmentasi *time stretch* dan *time shift*.

Model terbaik yang didapatkan kemudian akan diuji menggunakan data uji untuk mengetahui performa model pada data diluar data latih. Hasil pengujian ini disajikan dalam bentuk *confusion matrix* pada Gambar 6. Dari hasil tersebut dapat dihitung nilai *weighed average* akurasi, presisi, dan *f1-score* sebesar 95.59%, 95.65%, dan 95.6%. Performa model terbaik juga diuji pada Raspberry Pi 3B+, dimana durasi komputasi dan daya yang digunakan masing-masing sebesar 0.8 detik dan 2892 mW. Hasil tersebut tetap mengungguli performa yang dihasilkan dengan menggunakan *pretrained* VGG-16 pada Tabel 3.



Gambar 6 *Confusion matrix* model terbaik pada data uji

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, yaitu klasifikasi suara menggunakan *pretrained Convolutional Neural Network* (CNN) dapat disimpulakan bahwa penggunaan *pretrained* MobileNetV3-Small dan MLP memberikan performa yang paling baik untuk diimplementasikan pada *mobile device* seperti Raspberry Pi 3B+ dibandingkan kombinasi model lain yang diuji, dengan akurasi terbaik pada data latih tambahan dari augmentasi *time stretch* dan *time shift* sebesar 93.96% dan durasi *inference* pada Raspberry Pi 3B+ sebesar 0.8 detik. Performa klasifikasi dengan data augmentasi tersebut juga lebih tinggi dari pada data *original* dengan

peningkatan sebesar 2.9%. Selain itu, hasil dari penelitian ini menunjukan bahwa metode klasifikasi MLP lebih cocok untuk diimplementasikan dalam kasus ini, karena memiliki durasi komputasi yang paling cepat dan ukuran model yang tetap sama walaupun dengan data latih yang semakin banyak. Disisi lain, pengaruh *pretrained* VGG-16, MobileNetV3-Small, MobileNetV3-Large terhadap performa klasifikasi model tidak terlihat secara konsisten mengungguli satu dengan yang lain, sehingga dalam hal ini dapat disimpulkan bahwa ketiga *pretrained* tersebut memberikan performa klasifikasi yang sama. Akan tetapi, apabila dilihat dari waktu komputasi dan daya yang digunakan pada Raspberry Pi 3B+, terlihat bahwa *pretrained* MobileNetV3-Small memberikan hasil yang paling baik. Dari hasil tersebut dapat disimpulkan bahwa penggunaan *pretrained* MobileNetV3-Small memberikan performa yang lebih baik dari pada VGG-16 untuk klasifikasi suara pada *mobile device*.

DAFTAR PUSTAKA

[1] Sutoyo, "Keanekaragaman Hayati Indonesia Suatu Tinjauan : Masalah dan Pemecahannya," vol. 10, pp. 101–106, 2010.

[2] A. Damarraya and F. Bustomi Ahmad, *Deforestasi Indonesia Tahun 2019-2020*. 2021.

[3] Proffauna, "ProFauna's Report WILDLIFE TRADE SURVEY," 2009.

[4] A. Hamid and I. Amin, "Peranan Polisi Khusus Kehutanan Dalam Upaya Mencegah Dan Menanggulangi Penebangan Liar (Illegal Logging) Studi Di Kecamatan Moyo Hilir Kabupaten Sumbawa," *Ganec Swara*, vol. 15, no. 2, pp. 1266–1272, 2021.

[5] L. Nanni, G. Maguolo, S. Brahnam, and M. Paci, "An ensemble of convolutional neural networks for audio classification," *Appl. Sci.*, vol. 11, no. 13, pp. 1–27, 2021, doi: 10.3390/app11135796.

[6] S. Jagannathan, V. Sathiesh Kumar, and D. Meganathan, "Design and implementation of in-situ human-elephant conflict management system," *J. Intell. Fuzzy Syst.*, vol. 36, no. 3, pp. 2005–2013, 2019, doi: 10.3233/JIFS-169912.

[7] A. Howard, W. Wang, G. Chu, L. Chen, B. Chen, and M. Tan, "Searching for MobileNetV3," *Int. Conf. Comput. Vis.*, pp. 1314–1324, 2019.

[8] K. J. Piczak, "ESC: Dataset for environmental sound classification," *MM 2015 - Proc. 2015 ACM Multimed. Conf.*, pp. 1015–1018, 2015, doi: 10.1145/2733373.2806390.

[9] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," *MM 2014 - Proc. 2014 ACM Conf. Multimed.*, no. October, pp. 1041–1044, 2014, doi: 10.1145/2647868.2655045.

[10] S. Kahl *et al.*, "Overview of BirdCLEF 2021: Bird call identification in soundscape recordings," *CEUR Workshop Proc.*, vol. 2936, pp. 1437–1450, 2021.

[11] S. Guha, A. Das, P. K. Singh, A. Ahmadian, N. Senu, and R. Sarkar, "Hybrid feature selection method based on harmony search and naked mole-rat algorithms for spoken language identification from audio signals," *IEEE Access*, vol. 8, no. March 2021, pp. 182868–182887, 2020, doi: 10.1109/ACCESS.2020.3028121.

[12] S. Kahl, T. Wilhelm-Stein, H. Klinck, D. Kowerko, and M. Eibl, "Recognizing Birds from Sound - The 2018 BirdCLEF Baseline System," 2018, [Online]. Available: http://arxiv.org/abs/1804.07177.

[13] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, 2018, doi: 10.1007/s13244-018-0639-9.

[14] J. Geldmacher, C. Yerkes, and Y. Zhao, "Convolutional neural networks for feature extraction and automated target recognition in synthetic aperture radar images," *CEUR Workshop Proc.*, vol. 2819, pp. 86–91, 2020.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.