Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

Klasifikasi Suara Anjing Menggunakan Pretrained Model Yet Another Mobile Network Berbasis Convolutional Neural Network

Rich Deshan Djuardi*, Theresia Herlina Rochadiani

Fakultas Sains dan Teknologi, Informatika, Universitas Pradita, Tangerang, Indonesia
Email: [1,*]rich.deshan@student.pradita.ac.id, [2]theresia.herlina@pradita.ac.id
Email Penulis Korespondensi: rich.deshan@student.pradita.ac.id
Submitted: **11/05/2024**; Accepted: **23/06/2024**; Published**: 23/06/2024**

Abstrak−Dalam kehidupan sehari-hari, hewan peliharaan seperti anjing sering kali menjadi bagian tak terpisahkan dari kehidupan manusia. Motivasi untuk memelihara hewan peliharaan dapat bervariasi dari individu ke individu, mulai dari kebutuhan akan teman setia hingga tanggung jawab untuk merawat makhluk hidup lain. Di antara berbagai pilihan hewan peliharaan, anjing sering dianggap sebagai sahabat yang paling setia dan loyal terhadap manusia. Keunikan ini membuat banyak orang memilih untuk memelihara anjing sebagai bagian dari keluarga mereka. Seringkali, pemilik anjing mungkin tidak memahami pesan yang ingin disampaikan oleh suara yang dihasilkan oleh hewan kesayangan mereka. Suara-suara anjing ini memiliki tujuan khusus yang dapat mencerminkan berbagai emosi, seperti kegembiraan, kesedihan, atau kemarahan. Suara anjing juga dapat menjadi indikator kesehatan mereka yang perlu diperhatikan oleh pemilik. Fokus utama penelitian ini adalah untuk mengembangkan teknologi klasifikasi suara anjing untuk membantu para pemilik memahami dan berkomunikasi dengan anjing peliharaan mereka. Dalam penelitian ini, model YAMNet yang telah dilatih sebelumnya digunakan sebagai dasar untuk mengklasifikasikan berbagai peristiwa audio. Proses pelatihan model menggunakan algoritma CNN yang terdapat dalam arsitektur YAMNet. Total data yang digunakan berjumlah 373 data yang diklasifikasikan menjadi 4 kelas yaitu, *bark*, *howling*, *growling*, *whimper*. Hasil dari penelitian ini model mencapai akurasi 97,8% dengan nilai presisi, *recall*, dan *f1-score* untuk setiap kelas >= 95%.

Kata Kunci: Klasifikasi: Convolutional Neural Network; Anjing; Suara; YAMNet

Abstract−In everyday life, pets such as dogs often become an inseparable part of human life. Motivations for keeping a pet can vary from individual to individual, ranging from the need for a loyal companion to the responsibility of caring for another living creature. Among the various choices of pets, dogs are often considered the most loyal and loyal friends towards humans. This uniqueness makes many people choose to keep dogs as part of their family. Often, dog owners may not understand the message that the sounds produced by their beloved pets are trying to convey. These dog sounds have a special purpose that can reflect various emotions, such as joy, sadness, or anger. A dog's voice can also be an indicator of their health that owners need to pay attention to. The main focus of this research is to develop dog voice classification technology to help owners understand and communicate with their pet dogs. In this research, a pre-trained YAMNet model is used as a basis for classifying various audio events. The model training process uses the CNN algorithm contained in the YAMNet architecture. The total data used was 373 data which were classified into 4 classes, namely, bark, howling, growling, whimper. The results of this research model achieved 97.8% accuracy with precision, recall and f1-scores for each class >= 95%.

Keywords: Classification; Convolutional Neural Network; Dog; Sound; YAMNet

1. PENDAHULUAN

Dalam kehidupan sehari-hari, hewan peliharaan seperti anjing sering kali menjadi bagian tak terpisahkan dari kehidupan manusia. Motivasi untuk memelihara hewan peliharaan dapat bervariasi dari individu ke individu, mulai dari kebutuhan akan teman setia hingga tanggung jawab untuk merawat makhluk hidup lain. Berbagai pilihan hewan peliharaan, anjing sering dianggap sebagai sahabat yang paling setia dan loyal terhadap manusia. Keunikan ini membuat banyak orang memilih untuk memelihara anjing sebagai bagian dari keluarga mereka [1]

Anjing termasuk dalam kategori hewan peliharaan, yang seringkali menjadi teman dekat manusia. Oleh karena itu, tidak mengherankan jika banyak orang memilih anjing sebagai hewan kesayangan. Selain itu, sensitivitas yang tinggi dan beragamnya jenis anjing dengan penampilan menarik serta karakteristik yang unik masing-masing, menjadi alasan utama mengapa pemeliharaan anjing semakin populer. Karena itu, segala kebutuhan anjing diperhatikan dengan cermat untuk memastikan keberlangsungan hidup mereka [2].

Anjing sering mengeluarkan suara gonggongan untuk berinteraksi, mengekspresikan kecemasan, dan menunjukkan afeksi. Namun, pemilik anjing khususnya pemula terkadang tidak mengerti maksud dari suara yang dikeluarkan oleh anjing peliharaanya, gagasan mengenai gonggongan juga melibatkan banyak informasi fisik, yang merupakan metode yang aman, nyaman, dan efektif untuk berkomunikasi dan memperoleh informasi [3]. Pemahaman yang salah dan kurangnya informasi dalam memahami suara anjing dapat mengakibatkan dampak negatif bagi anjing peliharaan. Pemahaman yang baik tentang suara anjing sangat penting karena dapat membantu dalam mendeteksi suara yang dihasilkan oleh anjing peliharaan, mencegah perilaku agresif sehingga meningkatkan kesejahteraan anjing peliharaan. Ketika anjing menggeram atau *growling* menunjukkan bahwa emosi anjing tersebut sedang marah atau bersiaga, misalnya seekor anjing yang menjaga makanan dari sejenisnya (menjaga makanan), ketika diancam oleh orang asing (mengancam), dan ketika bermain tarik tambang dengan pemiliknya (bermain) [4]. Ketika anjing mengeluarkan suara, sering kali pemilik anjing tidak mengerti maksud dari suara yang dihasilkan anjing kesayangan mereka. Suara-suara yang dihasilkan dari anjing memiliki maksud tertentu yang dapat berkaitan dengan emosi seperti bahagia, sedih, dan marah, suara yang dihasilkan oleh anjing peliharaan juga mewakilkan tingkat kesehatan mereka [5]. Ketika anjing sedang sakit, terkadang mereka mengeluarkan suara-suara tertentu yang menandakan bahwa

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

kesehatan mereka sedang terganggu. Oleh karena itu, dengan memanfaatkan teknologi kecerdasan buatan, suara anjing dapat diklasifikasikan dengan menggunakan *Convolutional Neural Network* (CNN) sehingga pemilik dapat mengetahui maksud dari suara yang dikeluarkan anjing tersebut.

CNN adalah salah satu teknik yang paling umum dan luas diakui untuk memproses dan mengenali pola visual dalam berbagai tugas klasifikasi gambar. Penggunaan CNN dalam klasifikasi gambar mencakup berbagai masalah, mulai dari mengidentifikasi rambu lalu lintas hingga menafsirkan tulisan tangan, bahkan sampai mendeteksi tumor otak. Saat ini, penerapan CNN telah merambah ke bidang klasifikasi suara, di mana salah satu aplikasi yang umum adalah mengklasifikasikan suara-suara lingkungan atau suara umum [5]. Data untuk klasifikasi suara tidak selalu besar dalam jumlahnya dan sering kali memiliki ketidakseimbangan proporsi antara jumlah datanya, yang dapat mempengaruhi akurasi klasifikasi. Salah satu cara untuk menyelesaikan persoalan tersebut adalah dengan menerapkan model *transfer learning* pada struktur arsitektur yang relevan. *Convolutional Neural Network* (CNN). Arsitektur CNN yang telah dilatih sebelumnya dengan *dataset* tertentu, yang juga dikenal sebagai model *transfer learning* atau *pretrained model*, dapat digunakan untuk melatih *dataset* lainnya. [6].

Dalam penelitian berjudul "Klasifikasi suara kucing dan anjing menggunakan *convolutional neural network*" yang dilakukan oleh Cecilia Tania Emanuella dan timnya, mereka memanfaatkan kecanggihan *Convolutional Neural Network* (CNN) untuk menyortir suara-suara dari kucing dan anjing. Melalui eksplorasi pada *Dataset Audio Cat and Dog*, sebanyak 277 file wav telah diklasifikasikan ke dalam dua kelompok. Hasilnya mengungkap akurasi hingga 93,8%, akurasi validasi yang diperoleh sebesar 87,7%, *f1-score* sebesar 88% untuk tahap validasi, dengan ROC curve mencapai 87,8% [7].

Fuad Hamdi Bahar dengan timnnya melakukan penelitian yang berjudul "Klasifikasi Suara Kucing dan Anjing Menggunakan LSTM-GRU dan ANN-BP". Penelitian ini menggunakan 277 data suara, terdiri dari 164 suara kucing dan 113 suara anjing, kemudian, data tersebut dipisahkan menjadi data pelatihan sebesar 80% dan data uji sebesar 20%.. Metode LSTM-GRU dan ANN-BP digunakan untuk membedakan suara kucing dan anjing. Hasil penelitian menunjukkan akurasi sebesar 92% dengan precision 0.91 dan recall 0.91 [8].

Dalam riset yang dipimpin oleh Al Waasiu dan timnya, Model *Artificial Neural Network Multi-perceptron* digunakan dengan penuh keahlian untuk memisahkan suara-suara kucing dan anjing yang berjudul "Klasifikasi Audio *Cats and Dogs* Menggunakan Model *Artifical Neural Network Multi-perceptron*". Melalui serangkaian penelitian yang mendalam, mereka berhasil mencapai prestasi gemilang dengan akurasi mencapai 88,04% pada *epoch* ke-100. *Precision* yang didapat sebesar 88,04% dengan sensitivitas sebesar 86%. Hasil prediksi menunjukkan keberhasilan model dalam membedakan suara kucing dan anjing Hasil prediksi yang tajam dari model tersebut membuktikan kepiawaian mereka dalam membedakan suara khas kucing dan anjing, menyoroti kemampuan luar biasa dari metode yang digunakan [9].

Dwi Astuti, dalam penelitiannya yang tercatat dalam jurnal INISTA tahun 2019 berjudul "Aplikasi Identifikasi Suara Hewan Menggunakan Metode *Mel-Frequency Cepstral Coefficients* (MFCC)", mengusulkan aplikasi canggih untuk identifikasi suara hewan. Dengan berbekal Metode *Mel-Frequency Cepstral Coefficients* (MFCC) dan serangkaian teknik unggulan lainnya seperti DTW, Power spectrum, HMM, dan *Vector quantization*, mereka berhasil meraih tingkat akurasi yang memukau dalam pengenalan suara binatang secara real-time. Penelitian ini tidak hanya menyoroti kecanggihan teknologi, tetapi juga memberikan sumbangan berharga dalam pengembangan sistem identifikasi suara yang canggih dan andal [10].

Penelitian sebelumnya berjudul "*Cat Sounds Classification with Convolutional Neural Network*" yang dilakukan oleh Ridi Ferdiana dan timnya dalam *International Journal on Electrical Engineering and Informatics* tahun 2021 menggunakan *Convolutional Neural Network* untuk mengklasifikasikan suara kucing. Model yang dikembangkan berhasil memprediksi suara kucing dengan tingkat keberhasilan sebesar 88,473254% pada tahap pelatihan tetapi pada data pengujian, Ridi Ferdiana dan timnya mendapatkan akurasi sebesar 70.80734% [5].

Yet Another Mobile Network (YAMNet) merupakan model yang telah di-*pre-train* untuk melakukan klasifikasi audio secara *real-time*. Model klasifikasi kejadian audio tersebut menggunakan gelombang audio sebagai *input* dan membuat prediksi independen untuk masing-masing dari 521 kejadian audio dalam ontologi *AudioSet*. Model ini menggunakan arsitektur MobileNet v1 dan telah dilatih menggunakan korpus *AudioSet*. Dengan menggunakan Raspberry Pi 3, mikrofon komersial, dan serangkaian balok *Bluetooth Low Energy* (BLE), sistem ini mampu mendeteksi potensi kejadian berbahaya [11].

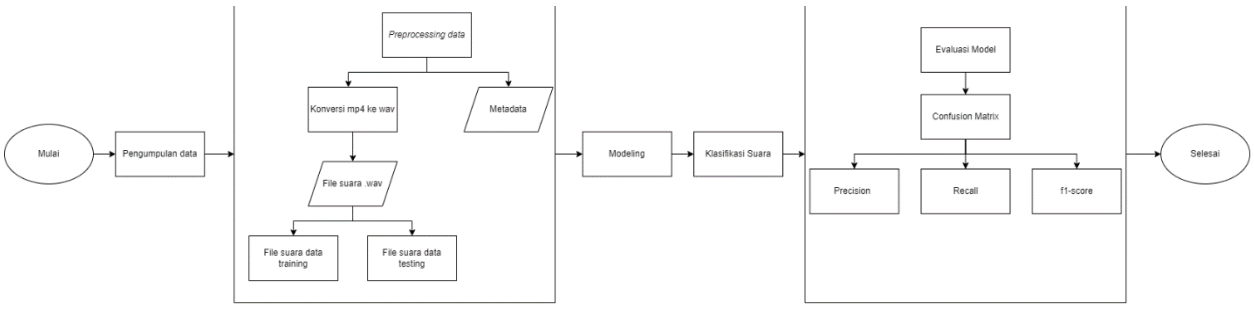
Meskipun sudah banyak penelitian mengenai klasifikasi suara hewan, belum ada penelitian yang secara khusus mengklasifikasikan maksud dari suara anjing, seperti gonggongan (*bark*), melolong (*howl*), menggeram (*growling*), dan gemetar (*whimper*). Penelitian ini bertujuan untuk menciptakan model yang mampu mengenali berbagai jenis suara anjing tersebut dengan menggunakan TensorFlow dan model YAMNet yang sudah di-*pretrained*. Selain itu, penelitian ini juga bertujuan untuk memperluas aplikasi pengenalan suara pada anjing guna mencakup berbagai jenis perilaku dan kondisi kesehatan yang mungkin terjadi pada hewan tersebut. Diharapkan penelitian ini dapat memberikan kontribusi bagi pengembangan lebih lanjut dan meningkatkan pemahaman tentang model klasifikasi suara anjing menggunakan *pretrained model* YAMNet.

2. METODOLOGI PENELITIAN

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

2.1 Tahapan Penelitian

Dalam studi ini, sejumlah langkah metodologis telah dilaksanakan dalam memperoleh data, mengevaluasi informasi, dan mencapai sasaran penelitian. Setiap langkah memiliki perannya sendiri dalam memastikan bahwa penelitian dilakukan secara sistematis dan dapat diandalkan, sejalan dengan maksud Klasifikasi Suara Anjing menggunakan YAMNet *Pretrained Model*. Tahapan penelitian dimulai dengan mengumpulkan data yang telah diambil dari internet, lalu melakukan *preprocessing data* agar data dapat diubah ke bentuk yang diinginkan.



Gambar 1. Tahapan Penelitian

Pada Gambar 1 *preprocessing data* dilakukan dengan mengkonversi *file* audio yang masih berbentuk mp4 menjadi *file* audio yang berbentuk wav. Setelah itu, dari kumpulan *file* wav yang sudah terkumpul dalam *folder* dibuat metadatanya menggunakan bahasa pemrograman python yang berisi nama *file* dan label. Label terdiri dari empat kelas yaitu *bark, whimper, growling,* dan *howl.* Lalu setelah *file* wav terbentuk, *file* wav digunakan untuk membagi *dataset* audio menjadi *dataset* untuk *training* dan *dataset* untuk *testing.* Setelah membagi *dataset*, dilakukan proses *modeling* yaitu melatih model dengan *dataset* dan selanjutnya menguji model menggunakan *dataset testing*. Setelah melakukan proses *testing*, klasifikasi suara dilakukan untuk mengetahui kelas yang terprediksi oleh model. Evaluasi model dilakukan setelah melakukan proses *testing* dengan menggunakan *confusion matrix*. Metrik *Precision, Recall,* dan *F1-score* digunakan untuk mengevaluasi kinerja model.

2.2 *Data Collection*

Pengumpulan data adalah proses mengumpulkan data dengan tujuan untuk mendapatkan wawasan mengenai topik penelitian [12]. Metode pengumpulan data yang diterapkan dalam penelitian ini yaitu *crawling*. *Crawling* merupakan suatu teknik otomatis untuk mengambil data dari sebuah situs web dengan menggunakan perangkat lunak komputer. Teknik ini memungkinkan pengambilan data dari berbagai sumber secara efisien dan terstruktur, yang berguna untuk berbagai keperluan seperti penelitian, analisis, dan pengembangan aplikasi [12]. Dalam penelitian ini, data diperoleh dari Google Audio Set melalui penggunaan tautan video YouTube yang selanjutnya diubah menjadi format audio dengan menggunakan bahasa pemrograman python untuk keperluan *deep learning*.

2.3 *Data Preprocessing*

Pra-pemrosesan data merupakan langkah awal yang penting dalam pengolahan data sebelum diterapkan pada model *machine learning*. Proses ini bertujuan untuk membersihkan, mentransformasi, dan mengorganisasi data secara tepat sehingga memungkinkan model *machine learning* untuk bekerja secara optimal. Salah satu aspek utama dari preprocessing data adalah peningkatan kualitas fitur, yang melibatkan berbagai teknik seperti pembuatan metadata, pengubahan format *file* audio, pengaturan titik awal audio dimulai dan panjang durasi audio dan lain sebagainya. Dengan melakukan preprocessing data secara tepat, kita dapat meningkatkan akurasi dan kinerja model *machine learning* serta menghindari masalah seperti *overfitting* atau *underfitting* [13].

2.4 *Data Splitting*

Pengelompokan data adalah ide yang merujuk pada pemisahan data menjadi dua kelompok atau lebih, yang sering disebut sebagai data pelatihan dan data pengujian. Konsep ini memegang peran krusial dalam bidang ilmu data, terutama dalam proses pembentukan model data. Dalam tahap pemisahan data menjadi dua bagian, umumnya terdapat data yang difungsikan sebagai latihan untuk mengembangkan model, sedangkan data uji digunakan setelah model selesai diperbarui. Proporsi pengelompokan data bergantung pada volume *dataset* yang tersedia, karena tidak ada standar atau kriteria yang pasti untuk pembagian data [14]. Berdasarkan *dataset* penulis, data yang memiliki nilai *fold* yang tidak habis dibagi 3 akan dimasukkan ke dalam *dataset* pelatihan (train_ds), sementara data dengan nilai *fold* habis dibagi 3 dan sisa 1 akan dimasukkan ke dalam *dataset* validasi (val_ds), dan data dengan nilai *fold* habis dibagi 3 dan sisa 2 akan dimasukkan ke dalam *dataset* pengujian (test_ds). Hal ini dilakukan agar menghindari data yang saling tumpang tindih pada saat data dibagi.

2.5 *Modeling*

Modeling adalah proses di mana model dari *machine learning* digunakan untuk menganalisis dan memahami pola-pola kompleks dalam data, dengan tujuan untuk memprediksi nilai-nilai yang tidak diketahui dari variabel-variabel

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

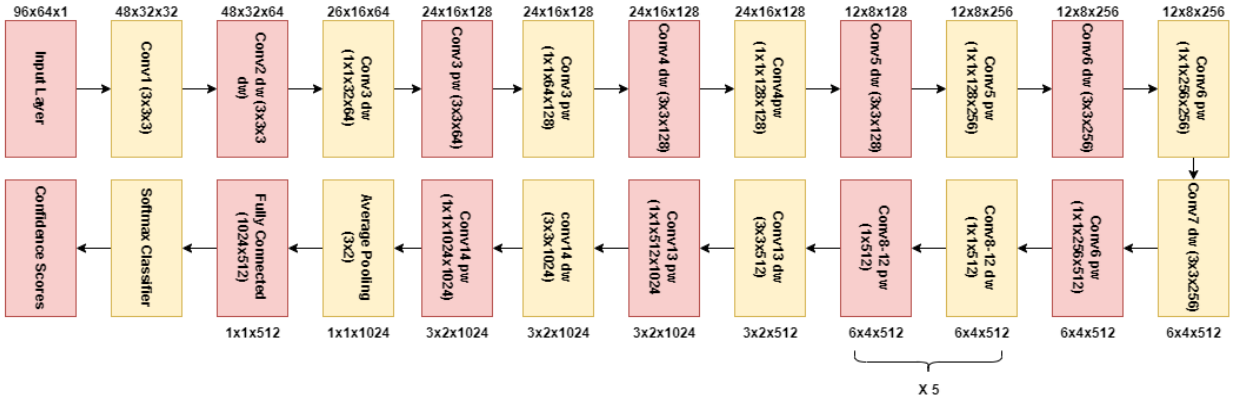
yang ada dalam data input. Proses ini melibatkan penggunaan algoritma-algoritma statistik dan matematika untuk membangun model yang dapat digunakan untuk melakukan prediksi berdasarkan data yang telah diamati sebelumnya, dengan menggunakan model ini, kita dapat mengembangkan pemahaman yang lebih baik tentang hubungan antara variabel-variabel *input* dan *output* dalam data, serta membuat estimasi atau prediksi tentang nilai-nilai yang mungkin dari variabel *output* yang belum diamati [15]. Pada penelitian ini, penulis menggunakan YAMNet *pretrained model* yang memiliki performa baik dalam mengklasifikasikan suara. Arsitektur YAMNet menggunakan algortima CNN yang cocok dalam melakukan proses *deep learning*.

2.5.1 YAMNet

Transfer learning merupakan salah satu aspek penting dalam pembelajaran mendalam di mana model dapat mempelajari fitur-fitur dari model lain yang telah dilatih sebelumnya. Aspek kunci dari *transfer learning* adalah untuk meminimalkan biaya komputasi dalam memanfaatkan pola-pola yang telah dipelajari sebelumnya. Penggunaan konsep *transfer learning* lebih disukai ketika terdapat sejumlah besar data tanpa label yang tersedia untuk melatih sebuah model. Oleh karena itu, model yang telah dilatih sebelumnya menggunakan fitur-fitur latihannya untuk mengurangi waktu dan usaha. YAMNet menggunakan MobileNetV1 sebagai jaringan dasar dan merupakan model yang telah dilatih sebelumnya pada *dataset* Google AudioSet untuk 521 peristiwa audio. Oleh karena itu, dalam penelitian kami, kami melatih YAMNet pada data yang tidak seimbang dan mencapai kinerja yang signifikan. Sebelum fase ekstraksi fitur, *resampling* dilakukan menjadi 16.000 Hz dengan audio satu saluran. Selain itu, YAMNet adalah model berbasis *deep learning* (DL), sehingga secara otomatis mengekstraksi fitur-fitur audio melalui lapisan ekstraksi fitur. Lapisan ekstraksi fitur mengekstraksi fitur-fitur audio dalam bentuk spektrogram, dan kemudian spektrogram ini dimasukkan ke dalam lapisan-lapisan MobileNet yang telah diperbaiki untuk klasifikasi [16].

2.5.2 *Training YAMNET Pretrained Model*

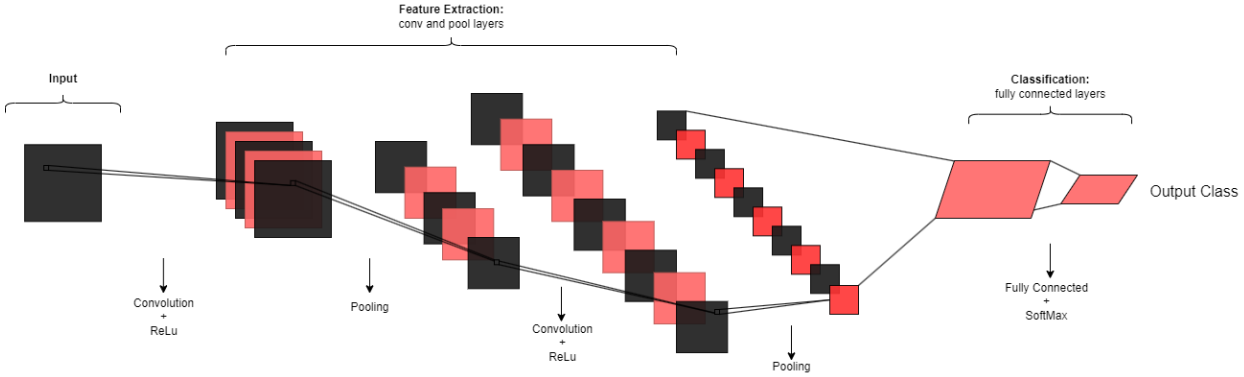
Data yang digunakan untuk melakukan proses *training* menggunakan YAMNet telah disesuaikan jumlahnya. Total *dataset* yang digunakan untuk memprosesnya berjumlah 373 audio yang terdiri dari 4 kelas, 249 audio digunakan untuk melakukan proses *training* (pelatihan), 125 audio digunakan untuk melakukan proses *validation* (validasi), dan 124 audio digunakan untuk *testing* (pengujian). Penelitian ini menggunakan arsitektur *Convolutional Neural Network* (CNN) yaitu YAMNet [17] .



Gambar 2. *Original* YAMNet *Architecture* [18]

Arsitektur YAMNet yang ditampilkan menggunakan jaringan dasar MobileNetV1 dan telah dilatih sebelumnya pada dataset Google AudioSet untuk mendeteksi 521 jenis peristiwa audio. Arsitektur ini dimulai dengan lapisan input yang menerima data audio dalam bentuk spektrogram berukuran 96x64x1. Lapisan ini diikuti oleh serangkaian lapisan konvolusi, dimulai dengan Conv1 yang menggunakan kernel 3x3 dan 32 filter untuk menghasilkan keluaran berukuran 48x32x32. Selanjutnya, terdapat lapisan Conv2 yang menggunakan *depthwise convolution* dengan kernel 3x3 dan 64 *filter*, menghasilkan keluaran berukuran 48x32x64, dan Conv3 dengan *pointwise convolution* yang memiliki kernel 1x1 dan 128 filter, menghasilkan keluaran berukuran 24x16x128. Lapisan ini diikuti oleh beberapa lapisan konvolusi tambahan dengan *depthwise* dan *pointwise convolution* yang berulang, menghasilkan keluaran dengan dimensi yang lebih kecil tetapi lebih dalam. Blok konvolusi ini diulangi sebanyak lima kali, menghasilkan keluaran akhir berukuran 6x4x512. Setelah itu, dilakukan *pooling* rata-rata untuk mereduksi dimensi menjadi 1x1x1024, diikuti oleh lapisan *fully connected* yang menghubungkan 1024 unit ke 512 unit. Akhirnya, lapisan *softmax* digunakan untuk menghasilkan skor kepercayaan untuk setiap kelas audio yang dideteksi. Arsitektur ini memungkinkan YAMNet untuk menangkap fitur-fitur penting dari data audio secara efisien melalui kombinasi lapisan konvolusi yang kompleks dan terstruktur. Selain itu, YAMNet secara otomatis mengekstraksi fitur audio dalam bentuk spektrogram yang kemudian diproses oleh lapisan MobileNet yang ditingkatkan untuk klasifikasi. Hal ini membuat YAMNet sangat efektif dalam mengklasifikasikan berbagai jenis suara dengan presisi tinggi [16].

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165



Gambar 3. Arsitektur CNN [19]

Pada Gambar 3, *Convolutional Neural network* (CNN) terdiri dari arsitektur berlapis dalam, di mana operasi aljabar multidimensi dilakukan secara berurutan. Arsitektur ini sangat cocok dengan data multidimensi karena informasi spasial dari data tersebut tetap terjaga selama pemrosesan. Misalnya, representasi aljabar dari gambar RGB membutuhkan dua dimensi untuk mengisi matriks piksel, dan dimensi ketiga untuk saluran RGB. Kemampuan CNN awalnya dimanfaatkan untuk tugas klasifikasi gambar, tetapi saat ini potensi arsitektur dalam ini telah diakui di berbagai bidang yang berkaitan dengan data multidimensi. Secara umum, CNN mencakup tiga jenis lapisan yang melakukan operasi spesifik lapisan konvolusi, lapisan pooling, dan lapisan *fully connected*. Lapisan *pooling* biasanya mengikuti lapisan konvolusi dalam arsitektur CNN. Keluaran dari lapisan *pooling* adalah peta fitur dengan resolusi rendah, yang diperoleh melalui *downsampling* peta input. Operasi *pooling* mengurangi jumlah parameter jaringan untuk mengurangi *overfitting* data. Selain itu, keluaran berdimensi rendah dari lapisan *pooling* mengekstraksi fitur lokal sambil mempertahankan multidimensionalitas data. Contoh *downsampling* adalah fungsi *Max Pooling*, yang hanya menyimpan nilai maksimum dari peta input di wilayah yang dilalui *filter*. Sebaliknya, *Global Average Pooling* melakukan rata-rata global pada peta *input*. Fitur yang dipelajari dari lapisan konvolusi dan *pooling* yang bertumpuk digabungkan dalam lapisan *fully connected* yang menerima vektor input yang telah diratakan. Jumlah *neuron* pada lapisan *fully connected* terakhir sesuai dengan jumlah kelas *output*. Untuk tugas klasifikasi yang umum, fungsi aktivasi pada lapisan *output* adalah fungsi saturasi seperti *Softmax*, yang dapat mengukur probabilitas kelas. Ketika data baru diberikan ke jaringan, fitur yang dipelajari akan diaktifkan dengan cara yang berbeda dan berujung pada lapisan *fully connected* terakhir untuk menentukan kelas output [19].

2.6 Evaluasi Kinerja Model

Confusion matrix adalah teknik yang seringkali dipakai untuk menghitung keakuratan proses penambangan data. *Confusion matrix* ditunjukkan melalui *table* yang memperlihatkan jumlah data pengujian yang berhasil diklasifikasikan dengan benar serta jumlah data pengujian yang salah pengklasifikasiannya [20]. *Confusion Matrix* digunakan dalam penelitian ini sebagai evaluasi kinerja model dalam memprediksi suara anjing pada saat pelatihan.

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

3. HASIL DAN PEMBAHASAN

Dalam penelitian yang dilakukan sekarang, model YAMNet yang telah dilatih sebelumnya digunakan sebagai dasar untuk mengklasifikasikan berbagai peristiwa audio. Proses pelatihan model melibatkan penggunaan *dataset* pelatihan (train_ds), *dataset* validasi (val_ds), *callbacks*, dan 20 *epochs* untuk mengatur berapa kali seluruh *dataset* pelatihan akan digunakan untuk melatih model. *Dataset* pelatihan digunakan untuk melatih model dengan pasangan *input-output* yang telah ditentukan, sementara *dataset* validasi digunakan untuk mengevaluasi kinerja model secara periodik selama proses pelatihan. Penggunaan *callbacks* memungkinkan intervensi dan penyesuaian model selama proses pelatihan, seperti penghentian pelatihan jika tidak ada peningkatan kinerja yang signifikan pada *dataset* validasi atau penyesuaian learning rate secara dinamis. Melalui proses ini, model YAMNet dapat disesuaikan dengan *dataset* yang spesifik dan menghasilkan kinerja yang optimal dalam mengklasifikasikan peristiwa audio. Dengan menetapkan jumlah *epochs* sebanyak 20, penelitian ini memungkinkan model untuk melatih dirinya dengan *dataset* tersebut selama periode yang ditentukan, mencapai tingkat akurasi yang diharapkan, dan mengoptimalkan performa klasifikasi suara anjing.

3.1 Pengumpulan data

Data yang dikumpulkan dalam penelitian ini dikumpulkan dari Google AudioSet. Google AudioSet menyediakan berbagai macam jenis-jenis suara, Google AudioSet menjadi pilihan penulis sebagai sumber untuk mengumpulkan *dataset* karena Google AudioSet menyediakan data suara dengan lengkap dan banyak, sehingga penulis bisa mengeksplor dataset yang penulis butuhkan.

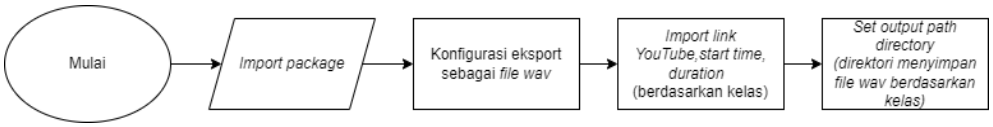
Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

Gambar 4. Diagram alir *data collection*

Dalam mengumpulkan data yang diperlukan dalam mengklasifikasikan suara anjing berdasarkan Gambar 4, menggunakan sumber *audioset* yang berasal dari Google Audio Set. Google Audio Set menyediakan berbagai macam kategori audio, tetapi yang diperlukan adalah *audioset* suara anjing. Google Audio Set menggunakan kumpulan video YouTube suara yang dihasilkan anjing dan menggabungkannya sesuai kategori dari video tersebut. Video tersebut diklasifikasikan lagi berdasarkan kategori suara yang dihasilkan oleh suara anjing. Kategori yang tersedia merupakan *bark*, *howl*, *growling*, *whimper*. *Bark* dikategorikan sebagai cara anjing berkomunikasi, *howl* dikategorikan sebagai pertanda bahwa anjing sedang kesepian. *Growling* dikategorikan sebagai marah atau keagresifan dan *whimper* dikategorikan sebagai merengek [5]. Google Colab (atau Google Colaboratory) adalah suatu *service cloud computing* yang diberikan oleh Google tanpa biaya. *Service* yang diberikan memberikan kebebasan terhadap pengguna menjalankan, program, menulisnya dan membagikan *code* tersebut kepada orang lain, serta melakukan pemrosesan data dan *machine learning*. Google Colab memberikan akses kepada pengguna untuk menggunakan sumber daya komputasi yang kuat tanpa memerlukan konfigurasi atau pengaturan khusus pada perangkat lokal. Selain itu, Google Colab juga menyediakan lingkungan pengembangan yang nyaman dan terintegrasi dengan alat-alat populer seperti Jupyter Notebook, mempermudah pengguna membuat dan menjalankan *notebook* interaktif dalam jaringan tanpa keharusan melakukan pemasangan perangkat lunak tambahan. Dengan demikian, Google Colab menjadi pilihan bagi penulis untuk melakukan proses *deep learning* untuk melakukan eksperimen dan mengembangkan proyek dengan mudah dan efisien [21].

3.2 Pra-proses data

Konversi format file audio dan video adalah tugas umum yang sering diperlukan untuk berbagai aplikasi, mulai dari analisis data audio hingga pengolahan multimedia. Salah satu konversi yang sering dilakukan adalah mengubah file video MP4 menjadi file audio WAV. File MP4 adalah format kontainer multimedia yang dapat menyimpan video, audio, dan teks, sedangkan file WAV adalah format audio yang digunakan untuk menyimpan rekaman suara dengan kualitas tinggi.



Gambar 5. Diagram alir konversi MP4 menjadi wav

Python satu dari bahasa pemrograman yang populer dan sering digunakan dalam membuat model *deep learning* [22]. Menyiapkan data sesuai dengan format yang diperlukan merupakan hal penting yang pertama kali dilakukan. berdasarkan Gambar 5, mengubah data yang awalnya merupakan format video (mp4) menjadi data yang berformat audio (wav). Beberapa komponen perlu disiapkan untuk mengubah format data seperti *URL*, *start_time*, dan *duration*. *URL* diperlukan untuk mendapatkan sumber dari video yang akan digunakan dan mengubahnya menjadi format audio. *Start_time* diperlukan untuk menentukan detik awal video yang akan dikonversi menjadi format audio. *Duration* merupakan durasi atau panjang video yang ditentukan di setiap video yang akan dikonversi menjadi audio, dalam kasus ini semua data memiliki *duration* selama 5 detik. Proses ini dilakukan untuk setiap kategori yaitu *bark*, *howl*, *growling*, dan *whimper*. *Howl* memiliki 109 data audio, *bark* memiliki 107 data audio, *growling* memiliki 103 data audio, dan *whimper* memiliki 54 data audio. Semua data audio dipilih berurutan dari Google Audio Set dan dikumpulkan sesuai kategorinya masing-masing. Kumpulan *URL* dikelompokkan untuk dimasukkan ke dalam folder kategori suara anjing tertentu, misalnya kumpulan *URL* yang berisikan suara-suara *bark* akan dikumpulkan dan nantinya hasil pengunduhan video menjadi audio tersebut akan langsung masuk ke dalam folder khusus audio berkategorikan *bark*, audio *howl* akan masuk ke dalam folder berkategori *howl*, audio *growling* akan masuk ke dalam folder berkategori *growling*, audio *whimper* akan masuk ke dalam folder berkategori *whimper*.

Table 1. Jumlah data setiap kelas

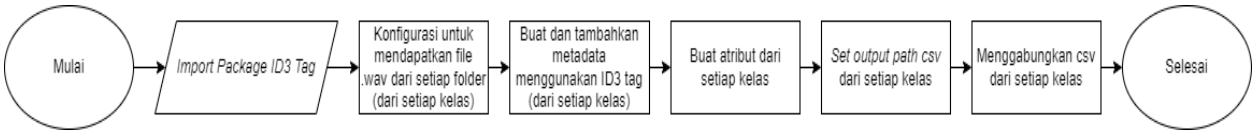
Class	Jumlah Data
Bark	107
Howl	109
Growling	103
Whimper	54

Tabel 1 menunjukkan distribusi jumlah data audio yang tersedia untuk setiap kelas suara anjing dalam dataset yang digunakan. Tabel ini mengklasifikasikan data audio ke dalam empat kategori utama: Bark, Howl, Growling, dan Whimper. Berikut adalah rincian jumlah data untuk masing-masing kelas:

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

a. *Bark*: Kategori ini mencakup 107 data audio. Bark biasanya merupakan suara yang dihasilkan anjing untuk berkomunikasi atau memperingatkan.
b. *Howl*: Kategori ini terdiri dari 109 data audio. Howl sering kali dikaitkan dengan anjing yang sedang kesepian atau mencoba berkomunikasi dengan jarak jauh.
c. *Growling*: Kategori ini memiliki 103 data audio. Growling biasanya menunjukkan anjing yang sedang marah atau merasa terancam.
d. *Whimper*: Kategori ini berisi 54 data audio. Whimper sering dihasilkan oleh anjing yang sedang merasa kesakitan atau merengek.

Dari tabel tersebut, dapat dilihat bahwa jumlah data terbanyak adalah untuk kategori Howl dengan 109 data, diikuti oleh Bark dengan 107 data, Growling dengan 103 data, dan yang paling sedikit adalah Whimper dengan 54 data. Distribusi data ini memberikan gambaran tentang seberapa banyak contoh suara yang tersedia untuk melatih dan menguji model klasifikasi suara anjing dalam penelitian ini.



Gambar 6. Diagram alir pembuatan metadata

Berdasarkan Gambar 6, proses pembuatan *model deep learning* menggunakan yamnet memerlukan metadata yang dibuat khusus yang merepresentasikan setiap data. Metadata dibuat berdasarkan format csv dan metadata dibuat menggunakan python dari setiap folder-folder yang sebelumnya berisi audio-audio dari setiap kategori suara anjing. Metadata ini memiliki kolom-kolom yang sudah ditentukan agar metadata ini dapat digunakan seperti *File Name*, label, *fold*, target. *File Name* berfungsi untuk menyimpan nama file audio, label berfungsi untuk menyimpan label atau kategori dari audio, *fold* berguna untuk menyimpan nilai numerik yang merupakan pembagian atau pengelompokan dari data untuk tujuan validasi silang atau pengujian model. Target berguna untuk menyimpan nilai target atau label yang diinginkan dalam proses klasifikasi atau prediksi. Setelah membuat metadata dari masing-masing folder, saatnya untuk menggabungkan metadata tersebut menjadi satu. Metadata yang tadinya dibuat berdasarkan masing-masing *folder*, kini digabungkan menjadi 1 metadata agar dapat digunakan dalam pembagian data dan pembuatan model.

3.3 Pembagian data

Pembagian data merupakan langkah krusial untuk memastikan bahwa model *deep learning* yang dikembangkan dapat dievaluasi secara akurat dan adil. Data yang telah dikumpulkan dibagi menjadi beberapa *subset* yang berbeda, yaitu data pelatihan (*training data*), data validasi (*validation data*), dan data pengujian (*testing data*). Pembagian ini dilakukan untuk memastikan bahwa model tidak hanya belajar dari data pelatihan tetapi juga mampu melakukan generalisasi pada data yang belum pernah dilihat sebelumnya. Pembagian data yang tepat sangat penting untuk menghindari *overfitting*, di mana model bekerja sangat baik pada data pelatihan tetapi gagal pada data pengujian karena tidak mampu menangkap pola yang lebih umum. Dengan membagi data menjadi *subset* yang berbeda, kita dapat memonitor performa model secara lebih menyeluruh dan melakukan penyesuaian yang diperlukan untuk meningkatkan kinerjanya. Data pelatihan digunakan untuk melatih model, data validasi digunakan untuk mengatur parameter model dan mencegah *overfitting*, sementara data pengujian digunakan sebagai benchmark akhir untuk menilai kemampuan prediksi model pada data yang benar-benar baru. Dengan demikian, proses pembagian data ini menjadi langkah fundamental yang menentukan keberhasilan model *deep learning* yang dibangun.

```
cached_ds = main_ds.cache()

# Filter data menjadi tiga fold
train_ds = cached_ds.filter(lambda embedding, label, fold: fold % 3 != 0)
val_ds = cached_ds.filter(lambda embedding, label, fold: fold % 3 == 1)
test_ds = cached_ds.filter(lambda embedding, label, fold: fold % 3 == 2)

# Hapus kolom fold karena tidak lagi diperlukan
remove_fold_column = lambda embedding, label, fold: (embedding, label)
train_ds = train_ds.map(remove_fold_column)
val_ds = val_ds.map(remove_fold_column)
test_ds = test_ds.map(remove_fold_column)

# Caching, pengacakan, pengelompokan, dan prefetching dataset
train_ds = train_ds.cache().shuffle(1000).batch(32).prefetch(tf.data.AUTOTUNE)
val_ds = val_ds.cache().batch(32).prefetch(tf.data.AUTOTUNE)
test_ds = test_ds.cache().batch(32).prefetch(tf.data.AUTOTUNE)
```

Gambar 7. *Data Splitting* (Pembagian Data)

Dalam proses yang ditunjukkan pada Gambar 7, langkah awal adalah menyimpan *dataset* utama ke dalam *cache* dengan perintah cached_ds = main_ds.cache(). Tindakan ini bertujuan untuk meningkatkan efisiensi pembacaan data dengan menyimpan data yang sudah dibaca ke dalam memori. Selanjutnya, *dataset* cached_ds disaring untuk

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

membagi data menjadi tiga bagian menggunakan filter() berdasarkan nilai *fold*. Data yang memiliki nilai *fold* yang tidak habis dibagi 3 akan dimasukkan ke dalam *dataset* pelatihan (train_ds), sementara data dengan nilai *fold* habis dibagi 3 dan sisa 1 akan dimasukkan ke dalam *dataset* validasi (val_ds), dan data dengan nilai *fold* habis dibagi 3 dan sisa 2 akan dimasukkan ke dalam *dataset* pengujian (test_ds). Setelah pembagian data, kolom fold tidak lagi diperlukan untuk proses pemodelan selanjutnya, sehingga fungsi remove_fold_column digunakan untuk menghapus kolom *fold* dari setiap data dalam *dataset*. Kemudian, fungsi map() diterapkan pada masing-masing *dataset* untuk menghapus kolom *fold*. *Dataset* pelatihan, validasi, dan pengujian kemudian disimpan ke dalam *cache*, dikelompokkan menjadi *batch* kecil dengan ukuran 32, dan dimuat ke dalam memori untuk mempercepat proses pembacaan data menggunakan fungsi cache(), batch(), dan prefetch() dengan parameter tf.data.AUTOTUNE**.** Kode ini menyediakan proses yang komprehensif untuk mempersiapkan data sebelum dilakukan pelatihan, validasi, dan pengujian model. Dengan membagi data menjadi tiga *fold*, menghapus kolom yang tidak diperlukan, dan melakukan *caching* serta pengelompokan data, proses pelatihan model dapat dilakukan dengan lebih efisien.

3.4 Pelatihan Model

Proses pelatihan menggunakan YAMNet *pretrained model* melibatkan penggunaan model jaringan saraf tiruan yang telah dipersiapkan sebelumnya dengan *dataset* yang luas dan beragam. Saat melakukan pelatihan, data yang diberikan dimasukkan ke dalam model untuk memperbarui *parameter* yang ada. Namun, karena YAMNet telah dilatih sebelumnya, proses pelatihan ini seringkali melibatkan metode *fine-tuning*, di mana parameter-parameter model disesuaikan sedikit agar lebih cocok dengan data yang spesifik yang dimiliki pengguna. Proses *fine-tuning* ini memungkinkan model untuk belajar pola dan fitur yang lebih khusus yang mungkin terdapat dalam *dataset* pengguna, sehingga meningkatkan kinerja dan akurasi model dalam tugas pengenalan suara. Selain itu, pelatihan menggunakan YAMNet pretrained model juga dapat melibatkan penggunaan teknik augmentasi data untuk meningkatkan keberagaman data pelatihan dan mencegah *overfitting*. Dengan memanfaatkan YAMNet *pretrained model* dalam proses pelatihan, penulis menghemat waktu dan sumber daya yang diperlukan untuk membangun model pengenalan suara dari awal, serta memperoleh hasil yang lebih konsisten dan akurat dalam waktu yang lebih singkat. Berikut merupakan gambar pelatihan yang dilakukan.

```
Epoch 1/20
78/78 [==============================] - 195s 2s/step - loss: 1.0003 - accuracy: 0.6667 - val_loss: 0.7760 - val_accuracy: 0.6851
Epoch 2/20
78/78 [==============================] - 2s 20ms/step - loss: 0.7222 - accuracy: 0.7246 - val_loss: 0.6669 - val_accuracy: 0.7282
Epoch 3/20
78/78 [==============================] - 1s 16ms/step - loss: 0.6508 - accuracy: 0.7521 - val_loss: 0.5642 - val_accuracy: 0.7828
Epoch 4/20
78/78 [==============================] - 1s 11ms/step - loss: 0.5955 - accuracy: 0.7736 - val_loss: 0.5318 - val_accuracy: 0.7958
Epoch 5/20
78/78 [==============================] - 1s 12ms/step - loss: 0.5448 - accuracy: 0.7861 - val_loss: 0.4685 - val_accuracy: 0.8275
Epoch 6/20
78/78 [==============================] - 1s 11ms/step - loss: 0.4779 - accuracy: 0.8202 - val_loss: 0.4783 - val_accuracy: 0.8405
Epoch 7/20
78/78 [==============================] - 1s 10ms/step - loss: 0.4597 - accuracy: 0.8343 - val_loss: 0.3931 - val_accuracy: 0.8723
Epoch 8/20
78/78 [==============================] - 1s 11ms/step - loss: 0.4049 - accuracy: 0.8582 - val_loss: 0.3892 - val_accuracy: 0.8682
Epoch 9/20
78/78 [==============================] - 1s 12ms/step - loss: 0.3830 - accuracy: 0.8631 - val_loss: 0.3092 - val_accuracy: 0.9032
Epoch 10/20
78/78 [==============================] - 1s 12ms/step - loss: 0.3312 - accuracy: 0.8866 - val_loss: 0.2784 - val_accuracy: 0.9121
Epoch 11/20
78/78 [==============================] - 1s 11ms/step - loss: 0.3019 - accuracy: 0.8996 - val_loss: 0.2396 - val_accuracy: 0.9276
Epoch 12/20
78/78 [==============================] - 1s 14ms/step - loss: 0.2806 - accuracy: 0.9109 - val_loss: 0.2035 - val_accuracy: 0.9422
Epoch 13/20
78/78 [==============================] - 1s 18ms/step - loss: 0.2343 - accuracy: 0.9202 - val_loss: 0.2004 - val_accuracy: 0.9390
Epoch 14/20
78/78 [==============================] - 1s 13ms/step - loss: 0.2240 - accuracy: 0.9287 - val_loss: 0.1754 - val_accuracy: 0.9447
Epoch 15/20
78/78 [==============================] - 1s 12ms/step - loss: 0.2074 - accuracy: 0.9348 - val_loss: 0.1616 - val_accuracy: 0.9561
Epoch 16/20
78/78 [==============================] - 1s 11ms/step - loss: 0.1863 - accuracy: 0.9429 - val_loss: 0.1380 - val_accuracy: 0.9593
Epoch 17/20
78/78 [==============================] - 1s 12ms/step - loss: 0.1650 - accuracy: 0.9453 - val_loss: 0.1237 - val_accuracy: 0.9658
Epoch 18/20
78/78 [==============================] - 1s 11ms/step - loss: 0.1480 - accuracy: 0.9559 - val_loss: 0.1058 - val_accuracy: 0.9715
Epoch 19/20
78/78 [==============================] - 1s 12ms/step - loss: 0.1441 - accuracy: 0.9534 - val_loss: 0.1033 - val_accuracy: 0.9748
Epoch 20/20
78/78 [==============================] - 1s 12ms/step - loss: 0.1234 - accuracy: 0.9587 - val_loss: 0.0877 - val_accuracy: 0.9780
```

Gambar 8. Hasil pelatihan data

Pada Gambar 8, kita melihat *output* dari pelatihan model menggunakan YAMNet *pretrained model* dalam 20 *epoch*. Setiap *epoch* mencakup 78 iterasi (*step*), di mana data diproses dan parameter model diperbarui. Selama proses pelatihan, kita dapat melihat perubahan dalam *loss* dan *accuracy* model pada setiap *epoch*, baik pada data pelatihan maupun data validasi. *Loss* adalah ukuran untuk mengukur berapa baik dan buruknya label yang benar, sedangkan *accuracy* adalah persentase prediksi yang tepat dibandingkan dengan total data. Dari *output* ini, kita dapat melihat bahwa dengan setiap *epoch*, *loss* pada kedua data pelatihan dan validasi cenderung menurun, sementara akurasi cenderung meningkat. Ini menunjukkan bahwa model sedang belajar pola-pola yang ada dalam data dan semakin baik

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

dalam memprediksi label. Pada akhirnya, model mencapai akurasi yang tinggi di atas 97% pada data validasi, menunjukkan kualitas model yang baik setelah melalui 20 *epoch* pelatihan.

3.5 Pengujian Data

Setelah melalkukan pelatihan, dilakukan proses pengujian data untuk mengetahui nilai akurasi yang dihasilkan oleh model terhadap data yang belum pernah ditemuinya. Proses ini penting dilakukan agar dapat mengetahui bahwa model dapat memprediksi data yang belum pernah diketahuinya dan tidak hanya bergantung pada data yang digunakan untuk melatihnya sehingga dapat diketahui bahwa model berhasil dalam melakukan proses pelatihan dan dapat memprediksi data baru.

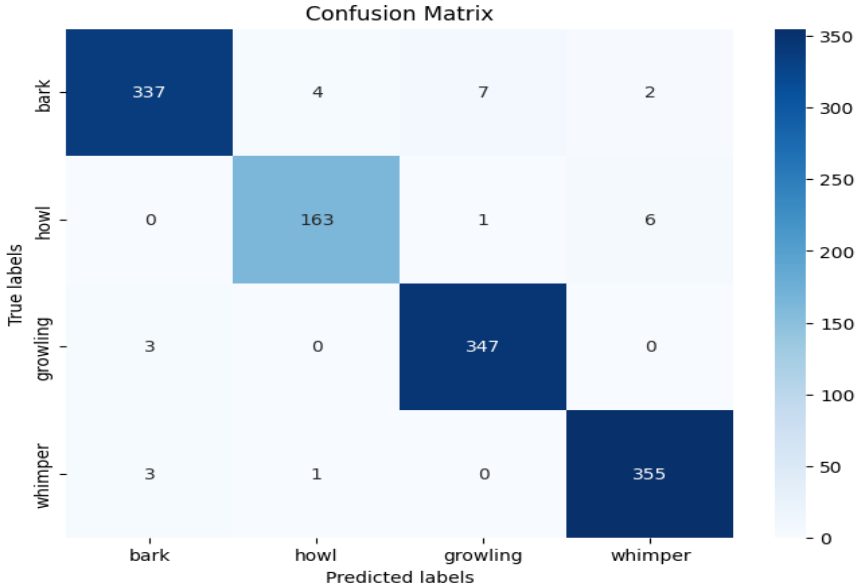
```
39/39 [==============================] - 0s 6ms/step - loss: 0.1170 - accuracy: 0.9645
Loss:  0.11700405180454254
Accuracy:  0.9645161032676697
```

Gambar 9. Akurasi data uji

Gambar 9 menunjukkan beberapa informasi penting yang berkaitan dengan evaluasi model. Pertama, kita memiliki informasi mengenai iterasi atau *step* yang dilakukan selama proses evaluasi, yaitu sebanyak 39 iterasi. Setiap iterasi ini melibatkan pengolahan sejumlah data yang telah ditentukan sebelumnya. Kemudian, terdapat informasi mengenai waktu yang diperlukan untuk melakukan setiap iterasi, yaitu 0.006 detik (6 milidetik) per iterasi. Selanjutnya, terdapat dua nilai yang menunjukkan kinerja model pada saat evaluasi, yaitu *loss* dan *accuracy*. *Loss* (kehilangan) adalah metrik yang mengukur seberapa akurat model dalam memprediksi label data. Pada kasus ini, nilai *loss* yang diperoleh adalah sebesar 0.1170. Semakin kecil nilai *loss*, semakin baik kinerja model. Selain itu, terdapat juga nilai *accuracy* (akurasi), yang merupakan persentase prediksi yang benar dibandingkan dengan total data yang dievaluasi. Pada kasus ini, nilai akurasi yang diperoleh adalah sebesar 96.45%, yang menunjukkan bahwa model mampu memprediksi dengan tepat sekitar 96.45% dari total data yang dievaluasi. Dengan demikian, informasi ini memberikan gambaran tentang seberapa baik kinerja model pada saat evaluasi dan seberapa akurat prediksi yang diberikan oleh model.

3.6 Evaluasi Model

Confusion matrix yang terlihat pada Gambar 10 merupakan suatu representasi visual berbentuk tabel yang berguna untuk mengevaluasi performa model klasifikasi. Tabel ini berisi 4 kelas yang berbeda yaitu: *bark*, *howl*, *growling*, dan *whimper*. Nilai-nilai dalam tabel tersebut menunjukkan jumlah prediksi yang dilakukan oleh model terhadap data uji.



Gambar 10. *Confusion Matrix*

Pada Gambar 10, *confusion matrix* menyediakan visualisasi dari hasil evaluasi model yang telah dibuat. Berikut ini adalah penjelasan lebih lanjut mengenai matriks tersebut:

a. Baris pertama menunjukkan data yang sebenarnya adalah "*bark*":
   1. Model memprediksi *bark* dengan benar sebanyak 337 kali (*true positive* untuk *bark*).
   2. Model salah memprediksi *bark* sebagai *howl* sebanyak 4 kali, sebagai *growling* sebanyak 7 kali, dan sebagai *whimper* sebanyak 2 kali.
b. Baris kedua menunjukkan data yang sebenarnya adalah *howl*:

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

1. Model memprediksi *howl* dengan benar sebanyak 163 kali (*true positive* untuk *howl*).
2. Model salah memprediksi *howl* sebagai "*bark*" sebanyak 0 kali, sebagai growlin sebanyak 1 kali, dan sebagai *whimper* sebanyak 6 kali.

c. Baris ketiga menunjukkan data yang sebenarnya adalah *growling*:
   1. Model memprediksi *growling* dengan benar sebanyak 347 kali (*true positive* untuk *growling*).
   2. Model salah memprediksi *growling* sebagai *bark* sebanyak 3 kali, sebagai *howl* sebanyak 0 kali, dan sebagai *whimper* sebanyak 0 kali.

d. Baris terakhir menunjukkan data yang sebenarnya adalah *whimper*:
   1. Model memprediksi *whimper* dengan benar sebanyak 355 kali (*true positive* untuk *whimper*).
   2. Model salah memprediksi *whimper* sebagai *bark* sebanyak 3 kali, sebagai *howl* sebanyak 1 kali, dan sebagai *growling* sebanyak 0 kali.

Warna yang lebih gelap pada sel-sel diagonal menunjukkan jumlah prediksi yang lebih tinggi yang benar, sementara warna yang lebih terang pada sel-sel lainnya menunjukkan jumlah prediksi yang salah atau *false positive* dan *false negative*. Sumbu vertikal (Y-axis) terdapat "*True labels*" yang menunjukkan kelas sebenarnya dari data, sedangkan pada sumbu horizontal (X-axis) terdapat "*Predicted labels*" yang menunjukkan kelas yang diprediksi oleh model. Di sebelah kanan matriks terdapat skala warna yang mengindikasikan jumlah data yang diprediksi dalam setiap sel matriks tersebut. Dalam evaluasi kinerja model, tercatat akurasi keseluruhan sebesar 97.80%. Akurasi adalah indikator yang mengukur persentase prediksi yang tepat dibandingkan dengan total data yang dinilai.

Table 2. Classification Report

Class	*Precision*	*Recall*	*F1 Score*
Bark	0.98	0.95	0.96
Howl	0.97	0.97	0.97
Growling	0.96	0.98	0.97
Whimper	0.98	0.99	0.98

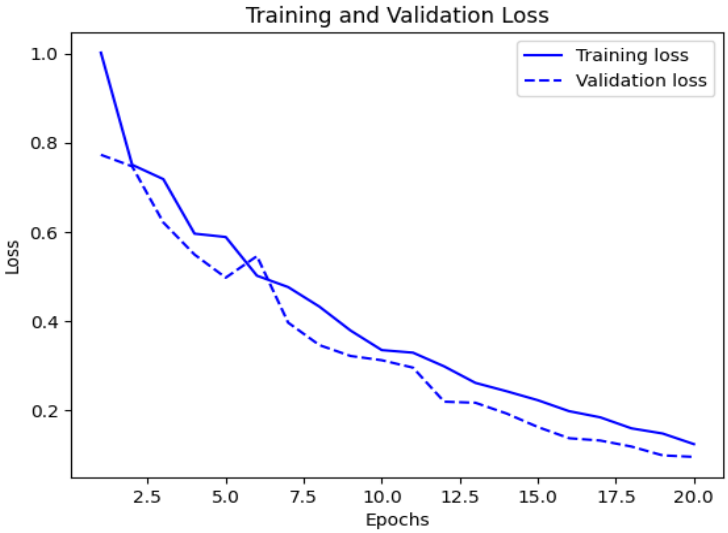
Metrik-metrik yang ditunjukkan pada Tabel 2 memberikan gambaran tentang kinerja model dalam memprediksi setiap kelas secara individual, dengan mempertimbangkan *trade-off* antara presisi dan *recall*. Presisi adalah ukuran untuk menilai jumlah prediksi positif yang benar, sementara itu, *recall* mengukur proporsi data positif yang berhasil diprediksi dengan tepat. *F1-Score* adalah ukuran gabungan dari kedua metrik tersebut. Dengan demikian, informasi ini membantu dalam mengevaluasi kekuatan dan kelemahan model dalam memprediksi masing-masing kelas. Nilai *precision, recall,* dan *f1*-score yang ditunjukkan Tabel 1 menunjukkan bahwa, kelas *bark* mendapatkan *precision* senilai 98%, *recall* sebesar 95%, dan *f1-score* sebesar 96%. Kelas *howl* mendapatkan *precision* sebesar 97%, *recall* sebesar 97%, dan *f1-score* yang juga 97%. Kelas *growling* memiliki *precision* dengan nilai 96%, *recall* sebesar 98% dan *f1-score* dengan nilai 97%. Terakhir, kelas *whimper* dengan nilai *precision* 98%, *recall* 99%, dan *f1-score* sebesar 98%. Data untuk kelas *whimper* memiliki jumlah data yang lebih sedikit dibandingkan dengan jumlah data pada kelas lainnya. Jumlah data yang lebih sedikit pada kelas *whimper* tetap memberikan nilai *precision, recall,* dan *f1-score* yang tinggi menyamakan nilainya dengan nilai *precision, recall, f1-score* dari kelas lain. Hal ini membuktikan bahwa dengan menggunakan *pretrained model* YAMNet, sangat membantu dalam mengatasi jumlah data yang tidak sebanding.

3.7 Kurva *Training and Validation Loss & Validation Accuracy*

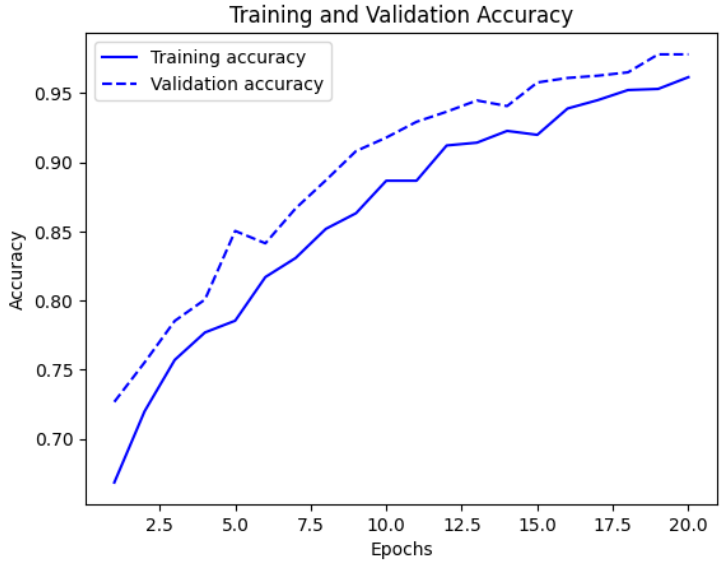
Pada Gambar 11 dan Gambar 12, menunjukkan grafik hasil pelatihan YAMNet *pretrained model* menunjukkan performa yang sangat baik dalam hal akurasi. Selama tahap pelatihan, awalnya model hanya mencapai akurasi sebesar 66,8%. Namun, seiring berjalannya proses training, dengan 20 *epoch* model berhasil mencapai akurasi sebesar 96%, menandakan kemampuannya yang sangat efektif dalam memprediksi data yang dipergunakan selama tahap pelatihan. Tingkat keakuratan yang tinggi dalam pelatihan ini menunjukkan bahwa model dapat mengidentifikasi pola dan fitur dalam data pelatihan dengan efisien. Selain itu, pada tahap validasi *loss*, pada *epoch* pertama model menunjukkan nilai validasi *loss* sebesar 77,2%. Model mengalami penurunan validasi *loss* hingga *epoch* ke-5 lalu validasi *loss* kembali mengalami peningkatan pada *epoch* ke-6 sebesar 54,5%. Setelah itu, model kembali mengalaim penurunan nilai hingga *epoch* ke-20, model berhasil mencapai nilai sebesar 9%. Tingkat validasi *loss* yang memadai mengindikasikan bahwa model juga dapat secara efektif beradaptasi dengan data baru yang tidak terdapat dalam *dataset* pelatihan. Kurva menunjukkan bahwa model mengalami *underfitting*, walaupun model mengalami *underfitting*, akurasi dan *loss* yang diraih masih sangat baik dalam mengklasifikasikan suara anjing. Dengan demikian, model ini mampu menghasilkan prediksi yang tepat pada data yang tidak dilibatkan dalam proses pelatihan. *Epoch* memainkan peran penting dalam menetapkan berapa kali model akan meninjau *dataset* pelatihan selama proses pembelajaran. Dalam kasus ini, 20 *epoch* memberikan model peluang yang memadai untuk mengidentifikasi dan menyesuaikan diri terhadap pola dan fitur yang ada dalam data pelatihan. Pada pelatihan ini, *shape* yang digunakan sebesar 1024 dan *dense layer* sebesar 512. Penggunaan dimensi 1024 dan dense layer 512 dalam YAMNet memberikan keuntungan yang signifikan dalam analisis audio. Dengan dimensi yang lebih besar, YAMNet dapat mengekstraksi fitur-fitur yang lebih kompleks dari data audio, memungkinkan pemahaman yang lebih dalam terhadap pola-pola yang rumit dalam variasi suara anjing. Hal ini juga meningkatkan kapasitas representasi model,

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

memungkinkan untuk merepresentasikan informasi yang lebih banyak dan lebih rumit dalam data. Dengan demikian, YAMNet dengan dimensi 1024 dan dense layer 512 memiliki potensi untuk memberikan hasil yang lebih baik dalam tugas klasifikasi suara, seperti mengklasifikasikan suara anjing, sambil tetap menjaga kemampuan generalisasi yang kuat.



Gambar 11. *Training and Validation Loss*



Gambar 12. *Training and Validation Accuracy*

Gambar 12 menunjukkan grafik akurasi pelatihan (*training accuracy*) dan akurasi validasi (*validation accuracy*) selama proses pelatihan model *deep learning* selama 20 *epoch*. Sumbu horizontal (X) mewakili jumlah epoch, yaitu iterasi penuh melalui dataset pelatihan. Grafik ini menunjukkan hasil dari 20 epoch pelatihan. Sumbu vertikal (Y) menunjukkan tingkat akurasi model, yang merupakan proporsi prediksi yang benar dibandingkan dengan total prediksi yang dibuat oleh model. Ditampilkan dengan garis *solid* berwarna biru. Grafik ini menunjukkan peningkatan yang konsisten dalam akurasi pelatihan seiring bertambahnya jumlah epoch. Akurasi pelatihan meningkat dari sekitar 70% di awal pelatihan hingga mencapai lebih dari 95% di akhir pelatihan. Ditampilkan dengan garis putus-putus berwarna biru. Akurasi validasi juga meningkat selama proses pelatihan, tetapi dengan pola yang sedikit berbeda dibandingkan akurasi pelatihan. Akurasi validasi mulai dari sekitar 70% dan meningkat hingga mendekati 95% pada akhir pelatihan. Kedua kurva menunjukkan bahwa model secara konsisten belajar dari data pelatihan dan juga mampu menggeneralisasi ke data yang belum pernah dilihat sebelumnya (data validasi). Hal ini ditunjukkan oleh peningkatan akurasi validasi yang sejalan dengan akurasi pelatihan. Kurva akurasi validasi yang berada di bawah kurva akurasi pelatihan menunjukkan bahwa meskipun model belajar dengan baik, masih terdapat sedikit perbedaan antara performa pada data pelatihan dan data validasi. Ini adalah hal yang wajar dalam pelatihan model machine learning. Secara keseluruhan, grafik ini menunjukkan bahwa model mengalami peningkatan performa yang signifikan selama proses pelatihan dan memiliki kemampuan yang baik untuk menggeneralisasi dari data pelatihan ke data validasi. Ini adalah

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

indikasi positif bahwa model yang dilatih memiliki performa yang baik dan tidak mengalami overfitting yang signifikan.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan tentang klasifikasi suara anjing menggunakan YAMNET *pretrained model* yang menggunakan 373 total data berbentuk audio, dapat disimpulkan bahwa YAMNet *pretrained model* merupakan model yang sangat baik dalam mengklasifikasikan suara anjing dibandingkan dengan menggunakan algoritma model dari penelitian penelitian sebelumnya. *Transfer learning* membantu model dalam mencapai akurasi yang optimal walaupun dalam *dataset* memiliki ketidakseimbangan jumlah data pada kelas *whimper* yang memiliki jumlah data lebih sedikir dibandingkan dengan jumlah data dari kelas lainnya. *Output* pelatihan model menggunakan YAMNet pretrained model selama 20 epoch menunjukkan peningkatan loss dan akurasi model pada setiap iterasi. Perubahan ini mengindikasikan bahwa model sedang memperbarui *parameter* dan belajar pola-pola dalam data. Akhirnya, model mencapai akurasi lebih dari 97% pada data validasi setelah 20 epoch pelatihan dan akurasi lebih dari 96% pada pengujian data, menunjukkan peningkatan kualitas model. Kurva *training and validation accuracy* serta kurva *training and validation loss* menunjukkan grafik yang sangat baik menunjukkan sedikit *underfitting* dengan nilai *gap* yang kecil yaitu sebesar 5,4% pada kurva *training and validation loss* dan 2,4% pada kurva *training and validation accuracy* pada saat proses pelatihan dan dapat disimpulkan proses pelatihan memberikan hasil yang akurat. YAMNet *pretrained model* merupakan metode yang sangat baik digunakan karena dengan 373 data berbentuk audio, YAMNet *pretrained model* dapat memberikan akurasi yang sangat signifikan. Masalah yang dialami penulis berada pada saat pengumpulan data yang merupakan suara asli anjing dalam kehidupan sehari-sehari, terkadang beberapa data memiliki *backsound* yang membuat kualitas suara anjing dari data tersebut menurun, tetapi YAMNet mampu memberikan tingkat akurasi yang tinggi dalam mengklasifikasikan suara anjing. Untuk meningkatkan performa pretrained model YAMNet yang mengalami *underfitting*, penulis menyarankan untuk mempertimbangkan beberapa langkah perbaikan. Pertama, penyesuaian arsitektur model dapat dilakukan dengan menambahkan lapisan-lapisan tambahan atau meningkatkan kompleksitas model agar dapat menangkap pola yang lebih kompleks dalam data. Selain itu, pemilihan *hyperparameter* yang optimal, seperti tingkat *learning rate* dan *batch size*, juga dapat membantu dalam meningkatkan performa model. Selain itu, memperluas *dataset* pelatihan dengan menambahkan lebih banyak data atau melakukan augmentasi data juga dapat membantu dalam meningkatkan generalisasi model. Terakhir, melakukan penalaan atau *fine-tuning* terhadap *layer* tertentu dari model YAMNet juga dapat menjadi langkah yang efektif untuk mengatasi *underfitting*.

REFERENCES

[1] M. Motamedi Fraser, “Dog words – or, How to think without language,” Sociol Rev, vol. 67, no. 2, p. 375, Mar. 2019, doi: 10.1177/0038026119830911.
[2] Muh. Amiruddin, I. G. N. Sudisma, and I. G. A. G. P. Pemayun, “UROLITHIASIS IN MALE POM MIX DOGS,” Veterinary Science and Medicine Journal, vol. 5, no. 12, p. 516, Jan. 2024, doi: 10.24843/vsmj.2023.v5.i12.p10.
[3] R. Deng, G. Zhou, L. Tang, C. Yang, and A. Chen, “E-DOCRNet: A multi-feature fusion network for dog bark identification,” Applied Acoustics, vol. 220, Apr. 2024, doi: 10.1016/j.apacoust.2024.109950.
[4] T. Faragó, N. Takács, Á. Miklósi, and P. Pongrácz, “Dog growls express various contextual and affective content for human listeners,” R Soc Open Sci, vol. 4, no. 5, p. 2, May 2017, doi: 10.1098/rsos.170134.
[5] R. Ferdiana, W. F. Dicka, and A. Boediman, “Cat Sounds Classification with Convolutional Neural Network,” International Journal on Electrical Engineering and Informatics, vol. 13, no. 3, p. 755, Sep. 2021, doi: 10.15676/ijeei.2021.13.3.15.
[6] A. J. Rozaqi, A. Sunyoto, and M. R. Arief, “Implementasi Transfer Learning pada Algoritma Convolutional Neural Network Untuk Identifikasi Penyakit Daun Kentang,” Seminar Nasional & Call Paper Fakultas Sains dan Teknologi, vol. 1, no. 1, Mar. 2021.
[7] C. T. Emanuella, M. Musfita, and A. Lawi, “Klasifikasi Suara Kucing dan Anjing Menggunakan Convolutional Neural Network,” Konferensi Nasional Ilmu Komputer (KONIK), vol. 5, no. 1, 2021.
[8] F. H. Bahar, N. I. Sari, and A. Lawi, “Klasifikasi Suara Kucing dan AnjingMenggunakan LSTM-GRU dan ANN-BP,” Konferensi Nasional Ilmu Komputer (KONIK), vol. 5, p. 202, Aug. 2021.
[9] E. M. Risondang, W. Andrean, F. P. Arieska, D. Astuti, and A. Junaidi, “Aplikasi Identifikasi Suara Hewan Menggunakan Metode Mel-Frequency Cepstral Coefficients (MFCC),” Journal of Informatics, Information System, Software Engineering and Applications (INISTA), vol. 1, no. 2, pp. 29–30, 2019.
[10] L. G. C. Vithakshana and W. G. D. M. Samankula, “IoT based animal classification system using convolutional neural network,” in 2020 International Research Conference on Smart Computing and Systems Engineering (SCSE), IEEE, Sep. 2020, pp. 90–95. doi: 10.1109/SCSE49731.2020.9313018.
[11] S. Bhosale, R. Chakraborty, and S. K. Kopparapu, “AUTOMATIC AUDIO CAPTIONING USING ATTENTION WEIGHTED EVENT BASED EMBEDDINGS,” Cornell University, 2022.
[12] H. Taherdoost, “Data Collection Methods and Tools for Research; A Step-by-Step Guide to Choose Data Collection Technique for Academic and Business Research Projects,” HAL Open Science, vol. 10, no. 1, pp. 10–38, 2021.
[13] O. Sami, Y. Elsheikh, and F. Almasalha, “The Role of Data Pre-processing Techniques in Improving Machine Learning Accuracy for Predicting Coronary Heart Disease,” International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, p. 812, 2021, doi: 10.14569/IJACSA.2021.0120695.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 30−42
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5165

[14] B. Supri, Rudianto, Abdurohim, Badriatul Mawadah, and Helmi Ali, "Asian Stock Index Price Prediction Analysis Using Comparison of Split Data Training and Data Testing," JEMSI (Jurnal Ekonomi, Manajemen, dan Akuntansi), vol. 9, no. 4, pp. 1403–1408, Aug. 2023, doi: 10.35870/jemsi.v9i4.1339.
[15] D. Wijaya and H. Jati, "PENGEMBANGAN MODEL DEEP LEARNING UNTUK PEMBANGKIT SOAL OTOMATIS MENGGUNAKAN RECURRENT NEURAL NETWORK," Jurnal Elektronik Pendidikan Teknik Informatika, vol. 10, no. 1, p. 37, 2022.
[16] R. Mahum, A. Irtaza, A. Javed, H. A. Mahmoud, and H. Hassan, "DeepDet: YAMNet with BottleNeck Attention Module (BAM) for TTS synthesis detection," EURASIP J Audio Speech Music Process, vol. 2024, no. 1, p. 5, Apr. 2024, doi: 10.1186/s13636-024-00335-9.
[17] A. Akram, K. Fayakun, and H. Ramza, "Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode Convolutional Neural Network," Building of Informatics, Technology and Science (BITS), vol. 5, no. 2, pp. 399–400, Sep. 2023, doi: 10.47065/bits.v5i2.4063.
[18] R. Mahum, A. Irtaza, A. Javed, H. A. Mahmoud, and H. Hassan, "Correction: DeepDet: YAMNet with BottleNeck Attention Module (BAM) for TTS synthesis detection," EURASIP J Audio Speech Music Process, vol. 2024, no. 1, p. 21, Apr. 2024, doi: 10.1186/s13636-024-00342-w.
[19] E. Brusa, C. Delprete, and L. G. Di Maggio, "Deep Transfer Learning for Machine Diagnosis: From Sound and Music Recognition to Bearing Fault Detection," Applied Sciences, vol. 11, no. 24, p. 11663, Dec. 2021, doi: 10.3390/app112411663.
[20] D. Y. Utami, E. Nurlelah, and F. N. Hasan, "Comparison of Neural Network Algorithms, Naive Bayes and Logistic Regression to predict diabetes," JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING, vol. 5, no. 1, pp. 53–64, Jul. 2021, doi: 10.31289/jite.v5i1.5201.
[21] Rangga Gelar Guntara, "Pelatihan Sains Data Bagi Pelaku UMKM di Kota Tasikmalaya Menggunakan Google Colab," Joong-Ki : Jurnal Pengabdian Masyarakat, vol. 2, no. 2, p. 246, Apr. 2023, doi: 10.56799/joongki.v2i2.1572.
[22] J. Hao and T. K. Ho, "Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language," Journal of Educational and Behavioral Statistics, vol. 44, no. 3, p. 5, Jun. 2019, doi: 10.3102/1076998619832248.