Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

Multi Kelas Speaker Recognition Menggunakan Deep Learning dengan CN-Celeb Dataset

Adipta Martulandi*, Amalia Zahra

Departement of Computer Science, Binus Graduate Program, Bina Nusantara University, Indonesia
Email: adipta.martulandi@binus.ac.id, amalia.zahra@binus.edu
Email Penulis Korespondensi: adipta.martulandi@binus.ac.id
Submitted: **02/11/2022**; Accepted: **07/12/2022**; Published**: 30/12/2022**

Abstrak−Speaker recognition sudah banyak diaplikasikan di berbagai bidang kehidupan manusia seperti Siri dari Apple dan Cortana dari Microsoft. Salah satu masalah saat pembuatan Speaker Recognition adalah terkait dataset yang digunakan untuk proses modeling. Dataset yang digunakan saat ini sebagian besar data yang belum bisa merepresentasikan kondisi nyata. Akibatnya adalah ketika diimplementasikan di dunia nyata hasilnya kurang maksimal. Penelitian ini mengembangkan model speaker recognition menggunakan deep learning (LSTM) dengan CN-Celeb dataset. CN-Celeb dataset adalah data yang diambil secara langsung dari dunia nyata sehingga banyak noise. Harapan menggunakan dataset ini adalah supaya bisa merepresentasikan kondisi dunia nyata. Penelitian ini Menggunakan 2 Stacked LSTM untuk multi-class speaker recognition task. Selain itu penelitian ini melakukan proses tuning hyperparameters dengan metode grid search untuk mendapatkan konfigurasi model yang paling optimal. Hasil penelitian menunjukan bahwa nilai EER model LSTM sebesar 10.13% lebih baik dari baseline paper acuan sebesar 19.05%. Selain itu, jika dibandingkan dengan penelitian lain yang juga menggunakan CN-Celeb dataset didapatkan bahwa model LSTM yang dibuat hasilnya lebih baik. Dari hasil penelitian yang telah dilakukan serta membandingkan dengan penelitian orang lain, didapatkan bahwa model LSTM yang dibangun memberikan performa yang lebih baik. Penelitian yang dibandingkan menggunakan model yang i-vectors, x-vectors, PLDA, LDA, dan transformers.

Kata Kunci: Deep Learning; Machine Learning; Pengenalan Pembicara; LSTM; CN-Celeb

Abstract−Speaker recognition has been widely applied in various fields of human life such as Siri from Apple, Cortana from Microsoft, and Voice Assistant by Google. One of the problems when creating speaker recognition is related to the dataset used for the modeling process. The dataset used for creating the speaker recognition model is mostly data that cannot represent real-world conditions. The result is when implemented in the real-world conditions are not optimal. This study develops a speaker recognition model using deep learning (LSTM) with the CN-Celeb dataset. The CN-Celeb dataset is data taken directly from the real world so there is a lot of noise. The hope of using this dataset is that it can represent real world conditions. Model development uses 2 stacked LSTM for multi-class speaker recognition tasks. In addition, this study performs tuning hyperparameters with a grid search method to obtain the most optimal model configuration. The results showed that the EER value of the LSTM model was 10.13% better than the reference baseline paper of 15.52%. In addition, when compared with other studies that also used the CN-Celeb dataset but using different models, it was found that the LSTM model had promising results. From the results of study that has been carried out and also compared with other people's research, it was found that the LSTM model gave promising performance. The LSTM model is compared with the x-vectors, PLDA, TDNN, and transformers models.

Keywords: Deep Learning; Machine Learning; Speaker Recognation; LSTM; CN-Celeb

1. PENDAHULUAN

Dengan berkembangnya teknologi komputasi secara tidak langsung berdampak pada Artificial Intelligence (AI) yang juga berkembang [1]. Berbagai penemuan di bidang AI telah banyak diterapkan dalam kehidupan manusia seperti pengenalan wajah pada smartphone yang termasuk dalam bidang computer vision, ada juga google translate yang termasuk dalam bidang pengolahan bahasa alami [2]. Selain itu, bidang teknologi speaker recognition juga mulai berkembang dengan berbagai implementasi pengenalan pembicara di bidang keamanan [3], rumah pintar [4], dan forensik [5].

Pengenalan pembicara atau speaker recognition adalah teknik dalam teknologi komputasi yang terdiri dari perangkat lunak dan sistem khusus yang dibuat untuk mengidentifikasi, membedakan, dan mengotentikasi suara seseorang [1]. Beberapa metode yang dapat digunakan untuk membuat model pengenalan speaker adalah menggunakan i-vector [6], x-vectors [7], PLDA [8], dan deep learning [9]. Selain metode, dataset juga perlu diperhatikan saat membuat model pengenalan speaker. Untuk saat ini beberapa dataset yang sering digunakan untuk membuat model pengenalan speaker masih belum dapat merepresentasikan kondisi nyata seperti dataset Vox-Celeb [6].

Hal ini dikarenakan dataset Vox-Celeb diambil dari video wawancara di YouTube yang artinya sebagian besar data sudah melalui proses editing dan cleansing. Itu sebabnya dataset Vox-Celeb tidak dapat merepresentasikan kondisi sebenarnya karena sebagian besar datanya bersih dan cenderung memberikan kinerja yang terlalu optimis [6]. Oleh karena itu, Fan et al [6] pada akhir tahun 2019 membuat sebuah dataset bernama CN-Celeb untuk mengatasi masalah tersebut. Sebagian besar data di CN-Celeb diambil langsung dari dunia nyata, sehingga banyak noise dan tidak bersih.

Fan et al [6] melakukan penelitian untuk membandingkan kinerja model yang dibangun dari dataset Vox-Celeb dan CN-Celeb. Hasil penelitian ini menunjukkan bahwa kinerja model yang dibangun dari data CN-Celeb tidak sebaik model dari Vox-Celeb dan penulis menyimpulkan bahwa data CN-Celeb lebih kompleks dan dapat lebih mewakili kondisi nyata. Sebagian besar penelitian yang dilakukan pada dataset CN-Celeb membahas tentang

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

normalisasi data dan tidak ada satupun yang berfokus pada model yang digunakan [7], [8]. Selain itu model yang digunakan dalam penelitian dataset CN-Celeb juga belum menggunakan deep learning khususnya LSTM [10]–[13].

Penelitian ini bertujuan untuk menutupi celah penelitian tersebut dengan melakukan penelitian pada dataset CN-Celeb untuk membuat model pengenalan speaker multi-kelas dengan harapan model yang dihasilkan mampu merepresentasikan kondisi nyata. Metode yang digunakan adalah LSTM dengan tujuan untuk memperkaya penelitian pada dataset CN-Celeb karena belum ada penelitian sebelumnya yang menggunakan LSTM. Untuk mendapatkan model yang optimal akan dilakukan tuning hyperparameter dengan menggunakan grid search.

Makalah ini disusun sebagai berikut. Bab 2 menjelaskan metodologi penelitian. Bab 3 melaporkan hasil percobaan dan membahas hasil. Akhirnya, Bab 4 menyimpulkan diskusi.

2. METODOLOGI PENELITIAN

2.1 Klasifikasi Menggunakan *Deep Learning*

Penelitian dan kajian terkait deep learning untuk pengenalan pembicara telah dilakukan dalam beberapa tahun terakhir. Model deep learning yang biasa digunakan adalah CNN dan RNN [14] [[15] [[16] [17]. Ketika peneliti menggunakan CNN, data audio biasanya diubah menjadi gambar spektogram karena CNN sangat baik untuk data gambar [14]. Penelitian lain menggunakan mesin limited boltzmann (RBM) dan mel frequency cepstral coefficient (MFCC) untuk ekstraksi ciri kemudian dimodelkan menggunakan CNN tanpa harus mengubah data audio menjadi citra spektogram [15].

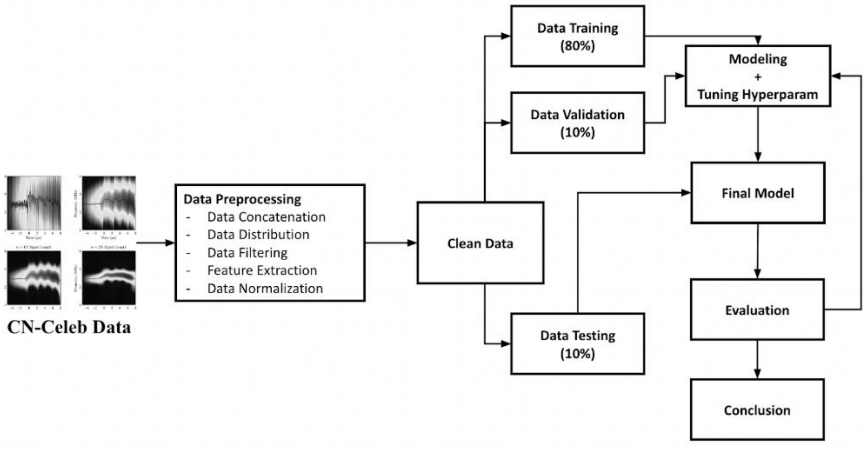
Penelitian dari saleem et al [5] membandingkan metode pengenalan pembicara tradisional seperti gaussian mix model-support vector machine (GMM-SVM) dengan metode deep learning deep neural network (DNN). Hasilnya adalah metode DNN memberikan kinerja yang lebih baik dibandingkan dengan metode tradisional. Pengenalan pembicara juga dapat diterapkan untuk menentukan jenis kelamin seseorang apakah dia perempuan atau laki-laki [16]. Setelah dilakukan percobaan, ternyata LSTM memberikan hasil yang paling optimal dibandingkan dengan KNN dan SVM untuk pengenalan gender. Dokuz et al [17] melakukan penelitian tentang penyetelan hyperparameters dengan LSTM untuk pengenalan speaker pada data korpus VCTK. Teknik penyetelan hiperparameter yang digunakan adalah pemilihan sampel minibatch yang hasilnya lebih baik dan dapat mengurangi kesalahan sebesar 5% dibandingkan tanpa teknik ini.

2.2 Pengenalan Pembiaca Menggunakan CN-Celeb Dataset

Dalam 3 tahun terakhir setelah dataset CN-Celeb dirilis ke publik, per hari ini telah diterbitkan sekitar 30 paper dan sebagian besar terkait dengan pengenalan pembicara dari data preprocessing hingga pengembangan model. Preprocessing data berfokus pada pengubahan hasil ekstraksi ciri agar terdistribusi secara normal [7], [8]. Lantian et al [7] menggunakan metode baru analisis diskriminan saraf (NDA) untuk mengubah distribusi vektor non-normal menjadi normal dan mengurangi EER sebesar 2%. Kemudian ada penelitian menggunakan metode lain yang disebut aliran normalisasi diskriminatif (DNF) yang fungsinya sama dengan NDA tetapi dengan pendekatan yang berbeda [8].

Selain itu, beberapa peneliti juga membahas model deep learning yang digunakan untuk dataset CN-Celeb [10] [18] [19]. Heo et al [10] menggunakan model yang basisnya adalah ECAPA-TDNN dan fitur yang digunakan adalah MFCC. Penelitian lain menggunakan metode pengenalan speaker tradisional yaitu x-vector yang digabungkan dengan DNF dan diperoleh nilai EER sebesar 12,13% [18]. Selanjutnya Ruiqi et al [19] melakukan penelitian dengan menggunakan model i-vector dan x-vector pada dataset CN-Celeb dan diperoleh nilai EER sebesar 16,59%. Namun, belum ada eksperimen pada dataset CN-Celeb menggunakan LSTM sebagai model.

2.3 Proses Penelitian



Gambar 1. Metode yang Diusulkan dan Flow dari Eksperimen

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

Studi kali ini akan menggunakan metode deep learning yaitu LSTM dengan menggunakan dataset CN-Celeb. Tujuan dari studi ini adalah membuat model speaker recognition untuk melakukan klasifikasi sebanyak 207 kelas. Proses eksperimen terdiri dari 4 tahapan yaitu data gathering, data pre-processing, modelling dan tuning hyperparameters, dan evaluation. Langkah pengerjaan selengkapnya sesuai dengan Gambar 1.

2.3.1 Data Gathering

Data yang digunakan dalam studi ini adalah CN-Celeb dataset yang dapat diunduh di website cnceleb.org. Dataset tersebut terdiri dari CN-Celeb 1 dan CN-Celeb 2, namun pada penelitian ini menggunakan data CN-Celeb 1. Data CN-Celeb 1 terdiri dari 997 speaker id dengan menggunakan bahasa china dan total audio file lebih dari 125.000. Beberapa challenges yang membuat dataset ini lebih kompleks dibandingkan dengan dataset yang sudah ada seperti Vox-Celeb adalah sebagai berikut.

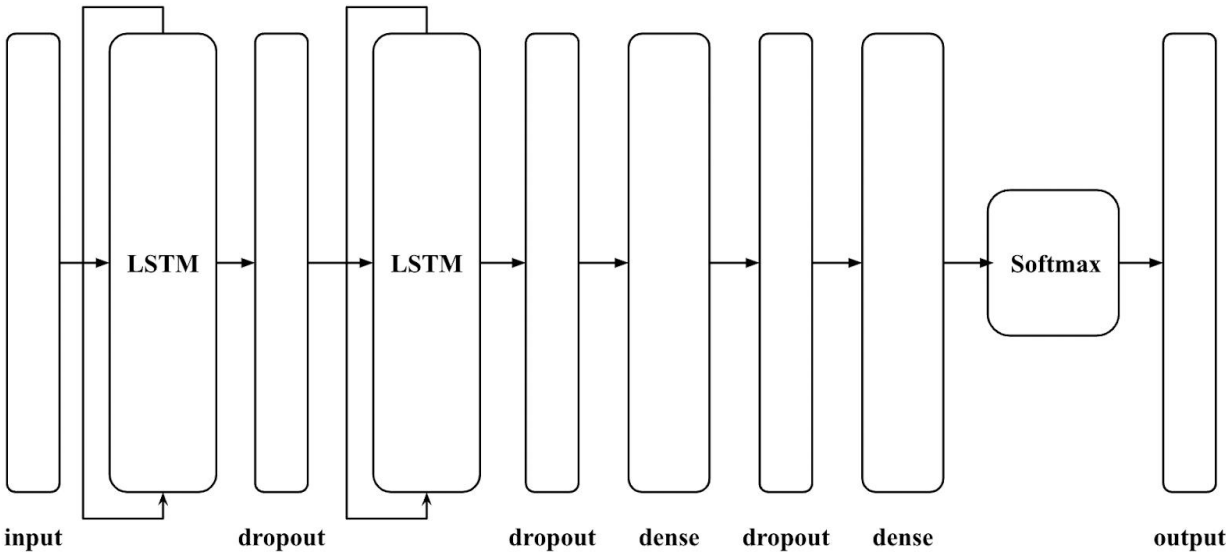
Beberapa audio file yang termasuk ke dalam genre movie dan drama memiliki overlapping background noise. Terdapat berbagai variasi di dalam speaking style karena 1 speaker bisa memiliki berbagai genre. Lalu 1 speakers bisa jadi direkam pada waktu dan devices yang berbeda yang bisa mengakibatkan cross-time and cross-channel problems. Sebagian besar audio terdapat real-world noise seperti ambient noise, background babbling, music, cheers and laugh [6].

2.3.2 Data Pre-Processing

Data CN-Celeb yang diambil langsung dari dunia nyata masih belum clean dan harus dilakukan beberapa preprocessing supaya datanya bersih [6]. Pada studi ini akan melakukan beberapa langkah untuk data preprocessing yaitu data concatenation, data distribution, data filtering, feature extraction dan data normalization.

2.3.3 Modelling dan Tuning

Proses modeling akan menggunakan data dengan durasi 10 detik dan testing akan dilakukan dengan data durasi 3 detik, 10 detik, dan 30 detik. Konfigurasi default dari LSTM adalah menggunakan 2 stack LSTM cell dengan jumlah block setiap cellnya adalah 100 [20]. Untuk activation function menggunakan reLU [21], optimizer menggunakan Adam [20], dan classifier menggunakan softmax [22] Untuk informasi lebih lanjut terkait arsitektur dari model bisa dilihat pada Gambar 2.



Gambar 2. Arsitektur Model Deep Learning yang Digunakan

2.3.4 Evaluation

Metode evaluasi yang bisa digunakan dalam penelitian terkait speaker recognition adalah equal error rate (EER) dan accuracy. EER adalah suatu nilai yang didapatkan ketika error tipe I atau false positive rate (FPR) nilainya sama dengan error tipe 2 atau false negative rate (FNR).

3. HASIL DAN PEMBAHASAN

Tahapan pertama dalam penelitian ini adalah pengumpulan dataset yang akan digunakan. Dataset yang akan digunakan adalah CN-Celeb yang dibuat dan dikembangkan oleh para peneliti dari Tsinghua University (http://www.openslr.org/82/). Untuk lebih detail terkait dataset bisa dilihat pada section 3.2

3.1 Pra - Pemprosesan

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

Tahap kedua adalah melakukan pra-pemrosesan yang prosesnya terdiri dari penggabungan data, pemotongan data, penyaringan data, ekstraksi fitur dan proses noise filtering yang dilakukan semuanya menggunakan bahasa pemrograman Python. Hasil akhir merupakan kumpulan fitur yang siap untuk melalui proses modeling.

3.2 Penggabungan Data

Setelah semua data terkumpul, hal yang pertama dilakukan adalah penggabungan semua data (concatenation) per speakers [5]. Jadi contohnya adalah ketika 1 speaker mempunyai 5 audio file dengan durasi 10 detik, 15 detik, dan 20 detik. Maka seluruh audio file dari speaker tersebut akan digabung menjadi 1 audio file besar dengan total durasi 45 detik. Ilustrasi dari proses penggabungan data dapat dilihat pada Gambar 3 di bawah.

Sebelum

No	Speaker Id	Audio File	Duration (s)
1	Id00001	audio_1.flac	16
2	Id00001	audio_2.flac	10
3	Id00001	audio_3.flac	3
4	Id00001	audio_4.flac	4
5	Id00001	audio_5.flac	8
6	Id00002	audio_1.flac	17
7	Id00002	audio_2.flac	23
8	Id00002	audio_3.flac	7
9	Id00002	audio_4.flac	12

Sesudah

No	Speaker Id	Audio File	Duration (s)
1	Id00001	audio_concat.flac	41
2	Id00002	audio_concat.flac	59

Gambar 3. Ilustrasi Penggabungan Data Sebelum dan Sesudah

Tujuan dari penggabungan data ini adalah untuk meminimalisir data yang terbuang karena semakin banyak data yang digunakan dalam proses pembuatan model LSTM kemungkinan besar akan meningkatkan performa dari model tersebut.

3.3 Pemotongan Data

Lalu proses selanjutnya adalah pemotongan data sesuai dengan durasi yang akan digunakan dalam penelitian. Penelitian ini akan menggunakan data dengan 3 durasi data yang berbeda yaitu 3 detik, 10 detik, dan 30 detik. Maksudnya adalah ketika 1 speakers data audio filenya sudah digabung dan memiliki 1 audio file dengan total durasi 45 detik, ketika data dipotong dengan durasi 3 detik maka akan didapatkan 15 audio file. Ketika dipotong dengan durasi 10 detik akan didapatkan 4 audio file dan ketika dipotong dengan durasi 30 detik akan didapatkan 1 audio file.
Jika ada data yang kurang dari durasi penelitian yaitu 3 detik, 10 detik, dan 30 detik maka akan dibuang data tersebut. Untuk proses training dalam penelitian ini akan menggunakan data dengan durasi 10 detik. Sedangkan untuk proses evaluasi akan menggunakan data dengan durasi 3 detik, 10 detik, dan 30 detik. Pemilihan untuk training model menggunakan data dengan durasi 10 detik adalah dengan mempertimbangkan dari sisi performa model dan jumlah data training yang digunakan. Ilustrasi proses pemotongan data dengan durasi 10 detik ada pada Gambar 4 dibawah, untuk proses pemotongan dengan durasi 3 detik dan 30 detik sama dengan 10 detik.

Sebelum

No	Speaker Id	Audio File	Duration (s)
1	Id00001	audio_concat.flac	41
2	Id00002	audio_concat.flac	59

Sesudah

No	Speaker Id	Audio File	Duration (s)
1	Id00001	audio_chunk_1.flac	10
2	Id00001	audio_chunk_2.flac	10
3	Id00001	audio_chunk_3.flac	10
4	Id00001	audio_chunk_4.flac	10
5	Id00002	audio_chunk_1.flac	10
6	Id00002	audio_chunk_2.flac	10
7	Id00002	audio_chunk_3.flac	10
8	Id00002	audio_chunk_4.flac	10
9	Id00002	audio_chunk_5.flac	10

Gambar 4. Proses Pemotongan Data Durasi 10 Detik

Ketika menggunakan data dengan durasi 3 detik untuk training model didapatkan jumlah data training yang cukup namun performa model yang rendah. Ketika menggunakan data dengan 30 detik, jumlah data training tidak

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

mencukupi dan performa model yang cukup. Seperti dengan proses sebelumnya, tujuan dari proses ini adalah untuk memaksimalkan jumlah data yang ada sehingga data yang terbuang akan diminimalisir.

3.4 Penyaringan Data

Dari proses 3.2 dan 3.3 didapatkan hasil speakers dan jumlah data training yang bisa digunakan. Karena jumlah datanya sangat bervariasi maka perlu kita samakan jumlah data untuk setiap speakers supaya tidak condong ke speakers dengan jumlah audio paling banyak. Pada penelitian ini hanya akan menggunakan speakers dengan jumlah data training minimal 150. Jadi jika 1 speakers memiliki jumlah data training sebanyak 300, maka hanya akan diambil top 150 audio file saja. Hal ini bertujuan supaya model LSTM belajar dari speakers dengan jumlah data yang sama (balance). Pemilihan jumlah data 150 karena dari hasil eksperimen menunjukan hasil yang bagus dan juga data yang digunakan juga tidak terlalu banyak berkurang. Sebelumnya sudah mencoba dengan jumlah 50 dan 100 namun hasilnya kurang maksimal. Setelah proses filtering, dari 997 speakers hanya ada 207 speakers yang akan digunakan dalam proses penelitian kali ini.

3.5 Feature Extraction Menggunakan MFCC

Proses selanjutnya setelah data training didapatkan adalah melakukan ekstraksi fitur pada data. Ekstraksi fitur yang akan digunakan adalah MFCC dengan nilai koefisien 13 [17] [23] [24]. Proses ekstraksi fitur menggunakan bahasa pemrograman Python dengan dibantu dengan Library Librosa. Secara lebih detail spesifikasi MFCC yang digunakan adalah sesuai dengan Tabel 1 di bawah.

Tabel 1. Spesifikasi MFCC

No	Variabel	Nilai
1	n_fft	2048
2	hop_length	512
3	sample rate	20250
4	n_mfcc	13

Proses ekstraksi fitur MFCC dilakukan 2x yaitu sebelum dan sesudah dilakukannya filtering noise hal ini bertujuan untuk melihat apakah dengan menggunakan filtering noise dapat meningkatkan performa model atau tidak. Filter yang digunakan adalah Wiener Filter, Low Pass, dan High Pass. Fitur MFCC yang telah diekstrak akan dilakukan proses perhitungan average, min, max, dan standard deviation. Tabel2 di bawah menunjukan contoh hasil ekstraksi fitur MFCC.

Tabel 2. Hasil Ekstraksi MFCC Koefisien 13

audio_file	mfcc_mean_1	mfcc_mean_2	mfcc_mean_3	mfcc_mean_4	mfcc_mean_5	mfcc_mean_6	mfcc_mean_7	mfcc_mean_8	mfcc_mean_9
1	-131.31	129.00	-60.78	41.51	-34.52	17.66	-23.95	4.66	-15.59
2	-159.17	128.17	-32.82	43.34	-29.19	21.09	-17.73	12.92	-6.01
3	-159.53	125.96	-35.14	47.66	-41.85	17.78	-17.88	7.11	-9.60
4	-154.72	128.35	-24.56	46.45	-28.16	23.19	-16.13	14.82	-3.98
5	-126.48	136.41	-26.10	57.43	-29.86	29.92	-18.01	16.44	-8.32

3.6 Pembagian Data Train, Validation, dan Test

Langkah selanjutnya adalah melakukan pemisahan data yang dibagi ke dalam train, validation, dan test dengan proporsi 80%, 10% dan 10%. Data train dan validation digunakan dalam proses tuning hyperparameters untuk mendapatkan nilai hyperparameters yang optimal untuk model deep learning. Sedangkan data test digunakan untuk menguji model deep learning dengan nilai hyperparameters yang optimal.

3.7 Standarisasi Data

Langkah terakhir sebelum proses modeling adalah melakukan standarisasi data. Standarisasi data digunakan untuk mengubah data ke dalam range 0 sampai dengan 1. Hal ini supaya proses training dengan LSTM lebih cepat dan mendapatkan hasil yang lebih baik. Metode standarisasi yang digunakan adalah Z-Score Normalization [21].

3.8 Proses Modeling dan Tuning Hyperparameters

Langkah Selanjutnya adalah proses modeling dan tuning hyperparameters. Proses modeling akan menggunakan data dengan durasi 10 detik dan testing akan dilakukan dengan data durasi 3 detik, 10 detik, dan 30 detik. Selain itu, akan ada 4 skenario dalam modeling yaitu modeling tanpa menggunakan noise filtering, menggunakan wiener filtering, menggunakan low pass filtering, dan menggunakan high pass filtering. Konfigurasi default dari model LSTM yang akan digunakan sesuai dengan Tabel 3.

Tabel 3. Konfigurasi Default LSTM

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

No	Tipe	Value
1	MFCC	13
2	Optimizer	Adam
3	LSTM Layer	2
4	Activation Function	reLU
5	Standard Scaler	StandardScaler
6	Classifier	softmax
7	:LSTM Unit	100
8	Dense Layer Unit	256

Lalu untuk proses tuning hyperparameters akan menggunakan 4 hyperparameters yaitu learning rate, batch size. epochs, dan dropout ratio. Secara detail nilai dari hyperparameters tersebut ada pada Tabel 4. Total kombinasi yang akan dijalankan dalam 1 proses modeling skenario adalah sebanyak 24 kali.

Tabel 4. Nilai Hyperparameters

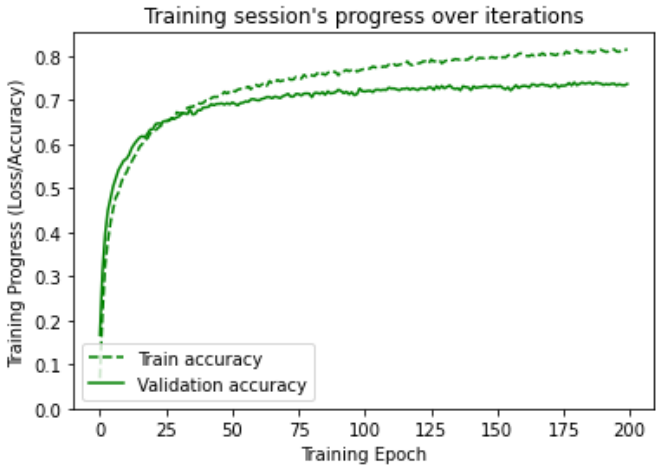
No	Type	Value
1	Learning Rate	[0.01, 0.001]
2	Batch Size	[128, 256]
3	Epochs	[100,150,200]
4	Dropout Ratio	[0.4, 0.5]

3.9 Eksperimen Durasi 10 detik Tanpa Filtering Noise

Pada eksperimen dengan durasi 10 detik tanpa filtering noise didapatkan bahwa jika melihat dari sisi metrics val_acc dan eer_mean dengan nilai 74,20% dan 12,95% didapatkan bahwa iterations eksperimen nomor 17 memberikan nilai yang paling optimal. Lalu setelah mendapatkan model eksperimen yang paling baik yaitu eksperimen 17, akan dilakukan testing menggunakan test dataset dengan durasi 3 detik, 10 detik, dan 30 detik sesuai dengan Tabel 5. Sedangkan untuk grafik training model tanpa noise filtering terbaik bisa dilihat pada Gambar 5 dibawah.

Tabel 5. Hasil Eksperimen Terbaik Durasi 10 Detik Tanpa Filtering Noise

iterations	filtering	Accuracy with Test Data Duration			EER with Test Data Duration		
		3s	10s	30s	3s	10s	30s
17	None	63.77%	79.82%	89.49%	18.20%	10.13%	5.28%



Gambar 5. Grafik Training Model Terbaik Tanpa Noise Filtering

3.10 Eksperimen Durasi 10 detik dengan Wiener Filtering Size 49

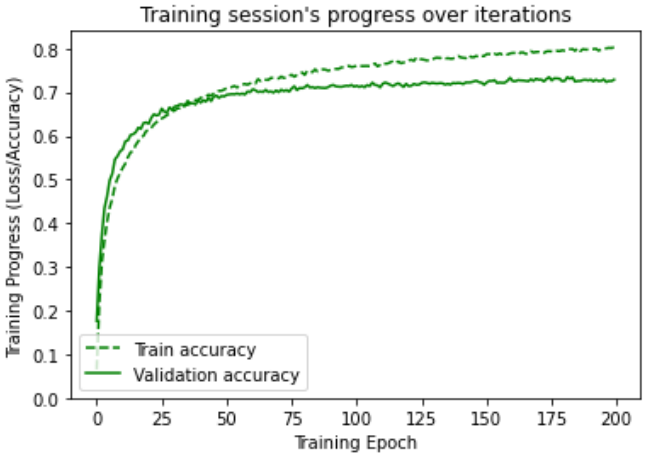
Pada eksperimen dengan Durasi 10 Detik dengan Wiener Filtering Noise didapatkan bahwa jika melihat dari sisi metrics val_acc dan eer_mean dengan nilai 72,90% dan 13,65% didapatkan bahwa iterations eksperimen nomor 17 memberikan nilai yang paling. Lalu setelah mendapatkan model eksperimen yang paling baik yaitu eksperimen 17, akan dilakukan testing menggunakan test dataset dengan durasi 3 detik, 10 detik, dan 30 detik. Hasil selengkapnya dari fase testing ada pada Tabel 6. Untuk grafik training model terbaik menggunakan wiener 49 noise filtering bisa dilihat pada Gambar 6 dibawah.

Tabel 6. Hasil Eksperimen Terbaik Durasi 10 Detik Dengan Wiener Filtering

iterations	filtering	Accuracy with Test Data Duration			EER with Test Data Duration		
		3s	10s	30s	3s	10s	30s

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

17	Wiener 49	61.29%	78.31%	88.95%	19.45%	10.91%	5.55%



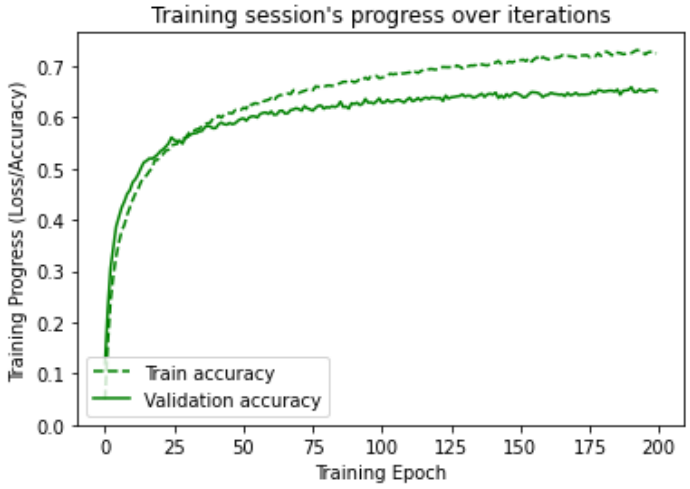
Gambar 6. Grafik Training Model Terbaik Wiener 49 Noise Filtering

3.11 Eksperimen Durasi 10 detik dengan Low Pass Filtering

Pada eksperimen dengan durasi 10 detik dengan Low Pass Filtering Noise didapatkan bahwa jika melihat dari sisi metrics val_acc dan eer_mean dengan nilai 64,60% dan 17,77% didapatkan bahwa iterations eksperimen nomor 17 memberikan nilai yang paling optimal. Lalu setelah mendapatkan model eksperimen yang paling baik yaitu eksperimen 17, akan dilakukan testing menggunakan test dataset dengan durasi 3 detik, 10 detik, dan 30 detik. Hasil selengkapnya dari fase testing ada pada Tabel 7. untuk grafik training model terbaik menggunakan low pass noise filtering bisa dilihat pada Gambar 7

Tabel 7. Hasil Eksperimen Terbaik Durasi 10 Detik Dengan Low Pass Filtering

iterations	filtering	Accuracy with Test Data Duration			EER with Test Data Duration		
		3s	10s	30s	3s	10s	30s
17	Low Pass	49.40%	71.27%	81.16%	25.43%	14.43%	9.47%



Gambar 7. Grafik Training Model Terbaik Low Pass Filtering

3.12 Eksperimen Durasi 10 detik dengan High Pass Filtering

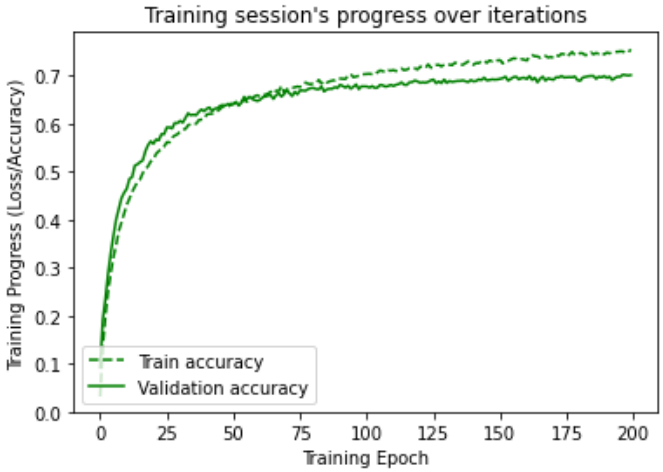
Pada eksperimen dengan durasi 10 detik dengan High Pass Filtering Noise didapatkan bahwa jika melihat dari sisi metrics val_acc dan eer_mean dengan nilai 64,60% dan 17,77% didapatkan bahwa iterations eksperimen nomor 23 memberikan nilai yang paling optimal. Lalu setelah mendapatkan model eksperimen yang paling baik yaitu eksperimen 23, akan dilakukan testing menggunakan test dataset dengan durasi 3 detik, 10 detik, dan 30 detik. Hasil selengkapnya dari fase testing ada pada Tabel 8. Untuk grafik training model terbaik menggunakan high pass noise filtering bisa dilihat pada Gambar 8.

Tabel 8. Hasil Eksperimen Terbaik Durasi 10 Detik Dengan High Pass Filtering

iterations	filtering	Accuracy with Test Data Duration			EER with Test Data Duration		
		3s	10s	30s	3s	10s	30s

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

23	High Pass	52.36%	74.40%	85.51%	23.94%	12.86%	7.28%



Gambar 8. Grafik Training Model Terbaik High Pass Filtering

Dari hasil eksperimen dengan 4 skenario diatas didapatkan bahwa untuk skenario 1, 2, dan 3 konfigurasi LSTM yang paling bagus adalah pada iterations 17. Konfigurasi 17 adalah ketika nilai learning rate sebesar 0,001, nilai batch size 128, nilai epochs 200 dan nilai dropout ratio 0.4. Sedangkan untuk skenario 4 konfigurasi paling baik adalah pada iterations 23. Konfigurasi 23 adalah ketika nilai learning rate sebesar 0,001, nilai batch size 256, nilai epochs 200 dan nilai dropout ratio 0.4.

Jika melihat dari hasil metrics test accuracy dan equal error rate (eer) didapatkan bahwa data test dengan durasi 30 detik memberikan hasil yang paling optimal disusul data dengan durasi 10 detik dan 3 detik. Hasil yang mengejutkan datang dari model tanpa filtering noise karena memberikan hasil test yang lebih bagus dari model yang diberikan noise filtering untuk semua durasi. Walau jika dilihat secara detail, hasil test untuk model tanpa filtering noise dan model dengan wiener 49 filtering hanya selisih sekitar 1% untuk semua durasi baik untuk accuracy maupun equal error rate.

3.13 Pembahasan Efek Durasi Audio Terhadap Performa Model

4 eksperimen diatas menunjukan bahwa hasil eksperimen dan evaluasi model data dengan durasi 30 detik memberikan hasil yang lebih baik jika dibandingkan dengan data durasi 10 detik dan 3 detik. Sebagai informasi tambahan, model LSTM ditraining menggunakan data dengan durasi 10 detik. Dari sini dapat dilihat pola bahwa semakin panjang durasi data, maka akan semakin bagus hasil yang didapatkan.

Paper tersebut menjelaskan bahwa semakin pendek durasi dari data yang digunakan, maka nilai equal error rate akan semakin besar. Nilai EER dari model i-vector yang digunakan turun dari 3,48% menjadi 22,09% saat durasi data yang digunakan dipersingkat dari 150 detik menjadi 2 detik [25]. Salah satu penyebab kenapa performa turun ketika durasi data diperpendek adalah karena variasi di dalam parameters intra-speakers, maksudnya adalah setiap durasi akan memiliki parameters yang berbeda-beda dan karena data suara merupakan data sequential maka hal ini akan berpengaruh.

Penelitian lain membuat sebuah model deep learning menggunakan data training Vox-Celeb 2 dan diuji dengan data Vox-Celeb 1. Salah satu hasil dari Penelitian tersebut menjelaskan bahwa data dengan durasi lebih dari 4 detik memberikan hasil yang lebih bagus dibandingkan dengan data yang kurang dari 4 detik. Ketika durasi data yang digunakan adalah 2 detik didapatkan nilai EER sebesar 7,97% sedangkan ketika durasi data yang digunakan adalah 6 detik didapatkan EER sebesar 3,39%. Menurut paper tersebut dengan seiring bertambahnya temporal length, ada kemungkinan yang lebih tinggi untuk dapat mengenali sinyal dari speaker yang asli [26].

Jika dilihat fitur yang digunakan yaitu MFCC, perbedaan durasi juga ada perbedaan dari nilai MFCC. Dapat dilihat pada Tabel 9 terdapat perbedaan yang signifikan dari MFCC durasi 3 detik dan MFCC durasi 30 detik. Hampir semua sampel kolom menunjukan perbedaan yang signifikan kecuali kolom mfcc_mean_2 dan mfcc_mean_5. Karena nilai pada kolom yang digunakan berbeda, maka hasil modeling juga akan berbeda.

Tabel 9. Perbedaan MFCC 3 detik dan 30 Detik

audio_file	mfcc_mean_1	mfcc_mean_2	mfcc_mean_3	mfcc_mean_4	mfcc_mean_5
speaker_01_3s	-223.290	130.973	-46.079	27.579	-44.432
speaker_01_30s	-193.220	118.284	-46.432	18.108	-29.846

3.14 Pembahasan Efek Noise Reduction

Hasil eksperimen yang telah dilakukan menunjukan bahwa Model tanpa noise filtering memberikan hasil yang paling baik dibandingkan dengan model yang menggunakan noise filtering. Menurut analisis ada beberapa hal yang

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

mengakibatkan hasil model yang menggunakan noise filtering kurang bagus yaitu Data CN-Celeb memiliki variasi noise yang banyak yaitu mulai dari ambient noise, background babbling, music, cheers dan laugh [27]. Yang berarti dataset tidak hanya terdiri dari stationary noise saja namun juga non-stationary noise. Hal ini menyebabkan wiener filtering tidak bisa perform dengan baik karena wiener filtering biasanya digunakan untuk noise yang stasioner.

Menurut penelitian dari Cai, Li, Wang dan Abel [28] dan Lantian, Dong dan Thomas [29] menunjukan bahwa hasil ekstraksi fitur data tidak berdistribusi normal. Hal ini mengakibatkan noise filtering yang digunakan belum bisa bekerja optimal karena noise filtering yang digunakan mengharuskan vektor harus berdistribusi normal.

Data CN-Celeb terdiri dari genre yang bervariasi, hal ini mengakibatkan kriteria atau ciri-ciri data sangat bervariasi. Hal ini mengakibatkan ada genre yang frekuensi suara rendah dan ada juga yang frekuensi suara tinggi. Hal ini mengakibatkan low pass dan high pass filtering kurang bisa bekerja dengan baik.

Proses penggabungan data pada section 3.2 hanya mempertimbangkan speaker id. Belum mempertimbangkan genre dari data. Jadi ada kemungkinan data yang digabung di dalam 1 speaker memiliki 2 genre atau lebih misal movie dan interview. Hal ini mengakibatkan dalam 1 audio file yang digabung terdapat 2 genre yang berbeda.

Proses training dan testing masih belum mempertimbangkan terkait genre, jadi masih banyak kemungkinan terjadinya mismatch pada saat training dan testing. Contohnya adalah proses training menggunakan genre interview sedangkan data yang ditest genre drama dan yang terakhir adalah proses hyperparameters yang digunakan pada noise filtering teknik belum maksimal atau belum optimal.

3.15 Pembahasan Terhadap Penelitian Orang Lain

Hasil eksperimen terbaik yang telah dilakukan akan dibandingkan dengan penelitian orang lain yang telah dilakukan dengan menggunakan CN-Celeb Dataset. Disini penulis akan membandingkan dengan 5 paper dari penelitian orang lain yang telah dilakukan dan akan membandingkan beberapa aspek seperti model yang digunakan, treatment yang dilakukan, dan metrics yang akan digunakan adalah EER dan accuracy.

Metrics EER dan accuracy untuk nomor 1 yang dicantumkan pada Tabel 10 di bawah adalah tanpa menggunakan noise filtering karena menghasilkan performa yang paling baik. Selain itu, karena training model yang dilakukan di dalam penelitian ini menggunakan data dengan durasi 10 detik, maka performa model yang dicantumkan pada Tabel 10 adalah untuk data dengan durasi 10 detik.

Dari 5 paper penelitian yang dibandingkan, dari sisi model yang digunakan tidak ada yang menggunakan model LSTM seperti yang disajikan dalam penelitian ini. Sebagian besar menggunakan model sistem i-vectors dan x-vectors. Selain itu yang menarik disini adalah ada penelitian yang menggunakan transformers, walaupun transformers memang sebagian besar digunakan untuk data yang berupa text.

Lalu dari sisi treatment yang dilakukan dalam proses penelitian sebagian besar berfokus kepada bagaimana untuk membuat distribusi hasil ekstraksi fitur itu menjadi distribusi normal seperti penelitian nomer 4. Selain itu penelitian nomer 5 berfokus kepada loss function yang digunakan yang mereka sebut dengan diffluence loss.

Dari sisi metrics yang digunakan sebagian besar penelitian menggunakan metrics EER saja, berbeda dengan penelitian ini yang ditambah dengan metrics accuracy. Walaupun konfigurasi penelitian berbeda, nilai EER yang didapatkan di dalam penelitian ini cukup baik yaitu di angka 10,13%. Hanya satu penelitian yang nilai EER lebih bagus dari penelitian ini yaitu nomor 1. Penelitian tersebut melakukan treatment Attention di fase back-end atau fase klasifikasi, metode Attention merupakan metode inti yang digunakan dalam transformer.

Tabel 10. Perbandingan Dengan Penelitian Lain

No	Researchers	Model	Dataset	Treatment	EER
1	Zeng et al [13]	TDNN	CN-Celeb	Attention Back-End	8.93%
2	This Research	LSTM	CN-Celeb	-	10.13%
3	Heo et al [10]	Ecapa-TDNN	CN-Celeb	Dino-Framework	10.65%
4	Cai et al [12]	PLDA	CN-Celeb	Deep Generative	11.66%
5	Zhang et al [11]	Transformers	CN-Celeb	Diffluence Loss	12.54
6	Fan et al [6]	x-vectors	CN-Celeb	-	15.52%

4. KESIMPULAN

Berdasarkan eksperimen yang telah dilakukan pada section 4 dapat diambil beberapa kesimpulan model tanpa menggunakan noise filtering memberikan hasil yang paling optimal jika dibandingkan dengan model yang menggunakan noise filtering. Metrics val_accuracy model tersebut di angka 74,20% dan nilai eer 12,95%. Konfigurasi Hyperparameters yang menghasilkan performa paling optimal adalah pada iterasi ke 17 untuk model tanpa noise filtering dengan nilai learning rate sebesar 0,001, nilai batch size 128, nilai epochs 200 dan nilai dropout ratio 0.4. Hasil test menunjukan bahwa data test durasi 30 detik memberikan nilai test accuracy dan test eer paling baik disusul dengan durasi 10 detik dan data durasi 3 detik. Serta noise filtering yang digunakan yaitu wiener filtering, low pass filtering, dan high pass filtering masih belum memberikan hasil yang optimal.

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1202−1211
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2467

REFERENCES

[1] J. P. Campbell, "Speaker recognition: A tutorial," Proceedings of the IEEE, vol. 85, no. 9, pp. 1437–1462, 1997.
[2] H. W. Lin and M. Tegmark, "Criticality in formal languages and statistical physics," arXiv preprint arXiv:1606.06737, 2016.
[3] N. Singh, R. A. Khan, and R. Shree, "Applications of speaker recognition," Procedia Eng, vol. 38, pp. 3122–3126, 2012.
[4] H. Ai, W. Xia, and Q. Zhang, "Speaker recognition based on lightweight neural network for smart home solutions," in International Symposium on Cyberspace Safety and Security, 2019, pp. 421–431.
[5] S. Saleem, F. Subhan, N. Naseer, A. Bais, and A. Imtiaz, "Forensic speaker recognition: A new method based on extracting accent and language information from short utterances," Forensic Science International: Digital Investigation, vol. 34, p. 300982, 2020.
[6] Y. Fan et al., "Cn-celeb: a challenging chinese speaker recognition dataset," in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 7604–7608.
[7] L. Li, D. Wang, and T. F. Zheng, "Neural discriminant analysis for deep speaker embedding," arXiv preprint arXiv:2005.11905, 2020.
[8] Y. Cai, L. Li, A. Abel, X. Zhu, and D. Wang, "Deep normalization for speaker vectors," IEEE/ACM Trans Audio Speech Lang Process, vol. 29, pp. 733–744, 2020.
[9] F. Ye and J. Yang, "A deep neural network model for speaker identification," Applied Sciences, vol. 11, no. 8, p. 3603, 2021.
[10] H.-S. Heo, J. Jung, J. Kang, Y. Kwon, Y. J. Kim, and B.-J. L. abd J. S. Chung, "Self-supervised curriculum learning for speaker verification," arXiv preprint arXiv:2203.14525, 2022.
[11] N. Zhang, J. Wang, Z. Hong, C. Zhao, X. Qu, and J. Xiao, "DT-SV: A Transformer-based Time-domain Approach for Speaker Verification," arXiv preprint arXiv:2205.13249, 2022.
[12] Y. Cai and D. Wang, "Deep generative LDA," arXiv preprint arXiv:2010.16138, 2020.
[13] C. Zeng, X. Wang, E. Cooper, X. Miao, and J. Yamagishi, "Attention back-end for automatic speaker verification with multiple enrollment utterances," in ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 6717–6721.
[14] S. Kadyrov, C. Turan, A. Amirzhanov, and C. Ozdemir, "Speaker Recognition from Spectrogram Images," in 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST), 2021, pp. 1–4.
[15] S. Hourri, N. S. Nikolov, and J. Kharroubi, "A deep learning approach to integrate convolutional neural networks in speaker recognition," Int J Speech Technol, vol. 23, no. 3, pp. 615–623, 2020.
[16] F. Ertam, "An effective gender recognition approach using voice data via deeper LSTM networks," Applied Acoustics, vol. 156, pp. 351–358, 2019.
[17] Y. Dokuz and Z. Tufekci, "Mini-batch sample selection strategies for deep learning based speech recognition," Applied Acoustics, vol. 171, p. 107573, 2021.
[18] Y. Cai, L. Li, D. Wang, and A. Abel, "Deep Speaker Vector Normalization with Maximum Gaussianality Training," arXiv preprint arXiv:2010.16148, 2020.
[19] L. Li et al., "CN-Celeb: multi-genre speaker recognition," Speech Commun, vol. 137, pp. 77–91, 2022.
[20] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," arXiv preprint arXiv:1707.06799, 2017.
[21] Z. Qin, D. Kim, and T. Gedeon, "Rethinking softmax with cross-entropy: Neural network classifier as mutual information estimator," arXiv preprint arXiv:1911.10688, 2019.
[22] M. Vacher, B. Lecouteux, J. S. Romero, M. Ajili, F. Portet, and S. Rossato, "Speech and speaker recognition for home automation: Preliminary results," in 2015 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), 2015, pp. 1–10.
[23] K. Nugroho and E. Noersasongko, "Enhanced Indonesian ethnic speaker recognition using data augmentation deep neural network," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 7, pp. 4375–4384, 2022.
[24] R. O. Ogundokun, R. Maskeliūnas, and R. Damaševičius, "Human Posture Detection Using Image Augmentation and Hyperparameter-Optimized Transfer Learning Algorithms," Applied Sciences, vol. 12, no. 19, p. 10156, 2022.
[25] A. Poddar, M. Sahidullah, and G. Saha, "Speaker verification with short utterances: a review of challenges, trends and opportunities," IET Biom, vol. 7, no. 2, pp. 91–101, 2018.
[26] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 5791–5795.
[27] Y. Fan et al., "Cn-celeb: a challenging chinese speaker recognition dataset," in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 7604–7608.
[28] Y. Cai and D. Wang, "Deep generative LDA," arXiv preprint arXiv:2010.16138, 2020.
[29] L. Li, D. Wang, and T. F. Zheng, "Neural discriminant analysis for deep speaker embedding," arXiv preprint arXiv:2005.11905, 2020.