SAINSTECH
Jurnal Penelitian dan Pengkajian Sains dan Teknologi
Volume 33 No. 4
Desember 2023
SINTA 5

Perancangan Sistem Deteksi Dan Pengenalan Perintah Suara Menggunakan Modul Esp 32 Dengan Metode Convolutional Neural Network (CNN)

Harlan Effendi[1], Rezky Mahardika Saryadi[2]

Program Studi Teknik Elektro, Fakultas Teknologi Industri, Institut Sains dan Teknologi Nasional
Jl. Moh Kahfi II, Srengseng Sawah, Jagakarsa, Jaksel 12630 telp.(021)7270090
Email : harlan@istn.ac.id, rezk002@brin.go.id

ABSTRAK

Kemampuan mikrokontroler pada saat ini berkembang sangat pesat. Berkat Hukum Moore jumlah transistor yang tertanam bertumbuh secara eksponensial. Hal ini menyebabkan bertambah canggihnya kemampuan mikrokontroler yang berbanding terbalik dengan harga, sehingga saat ini kita dapat menanamkan kecerdasan buatan pada mikrokontroler dengan salah satu bantuan API google yaitu TensorFlow lite. Pada penelitian tugas akhir ini akan dirancang sistem deteksi dan pengenalan perintah suara menggunakan mikrokontroller 32 bit berupa modul esp32 dengan kemampuan konektivitas wifi sebagai pemroses utama untuk melakukan pengolahan dan pengenalan data berupa perintah suara, pengambilan masukkan suara dilakukan mikrofon dengan teknologi MEMS (*Micro Electro Mechanical System*) menggunakan antarmuka I2C . Data yang dikirim memiliki lebar 8 bit dengan frekuensi sampling sebesar 44 Khz, hasil data sampling akan digunakan pada proses pelatihan dan pengujian kecerdasan buatan ditambah dengan kumpulan pustaka perintah suara suara dengan besar data sebesar 4 GB yang terdiri dari 20 file kata dalam bahasa inggris, bahasa indonesia dan sampel suara *noise* atau latar belakang lingkungan. Pelatihan dan pengujian dilakukan dengan mengubah hasil data sampling sinyal suara menjadi citra spektrum suara untuk masukkan pada algoritma CNN (*Convolutional Neural Network*. Hasil percobaan diperoleh proses pengenalan perintah suara memiliki tingkat ketepatan mencapai 90 % dengan waktu pengenalan kurang dari 1 detik.

Kata Kunci : esp32,Suara, CNN, MEMS, tensorflow Lite

ABSTRACT

The capabilities of microcontrollers are currently developing rapidly. Thanks to Moore's Law, the number of embedded transistors is growing exponentially. This has led to increasingly sophisticated microcontroller capabilities that are inversely proportional to price, so that we can now embed artificial intelligence on microcontrollers with the help of one of Google's APIs, TensorFlow Lite. In this final project research, a voice command detection and recognition system will be designed using a 32-bit microcontroller in the form of an ESP32 module with WiFi connectivity as the main processor to perform data processing and recognition in the form of voice commands. Voice input is taken using a MEMS (Micro Electro Mechanical System) microphone using the I2C interface. The data sent has a width of 8 bits with a sampling frequency of 44 kHz. The sampling data results will be used in the artificial intelligence training and testing process, together with a voice command library collection with a data size of 4 GB consisting of 20 word files in English, Indonesian, and noise or environmental background samples. Training and testing are carried out by converting the results of the voice signal sampling data into voice spectrum images for input to the CNN (Convolutional Neural Network) algorithm. The results of the experiment show that the voice command recognition process has an accuracy of 90% with a recognition time of less than 1 second

Keyword : esp32,Voice, CNN, MEMS, tensorflow Lite

DOI: https://doi.org/10.37277/stch.v33i4.1781

1. PENDAHULUAN

Pengenalan ucapan otomatis adalah metode menerjemahkan sinyal suara menjadi rangkaian kata menggunakan program komputer dan algoritmanya. Kemampuan komputer untuk mengidentifikasi ucapan "menerima dan menafsirkan" dan menerjemahkannya ke dalam bentuk atau teks yang dapat dibaca dikenal sebagai pengenalan ucapan otomatis. Pengenalan ucapan otomatis atau selanjutnya kita sebut sebagai Autonmatic Speech Recognition (ASR) merupakan kemampuan komputer untuk memahami ucapan serta melakukan tindakan berdasarkan instruksi manusia .

Sehingga diperlukan metode tertentu dalam pengolahan pengenalan ucapan bertujuan untuk mampu mengenaali ucapan dalam kondisi noise tinggi atau lingkungan yang tidak mendukung. Salah satu metode yang diperkenalan pada akhir tahun 1960 adalah metode Hidden Markov Model, metode ini berupa model statistika dari rantai markov, dengan metode HMM proses real time secara umum menghasilkan observable output yang dapat dikarakterisasikan sebagai sinyaal. Sinyal bisa bersifat diskrit (karakter dalam alfabet) maupun kontinu (pengukuran temperatur, alunan musik). Sinyal bisa bersifat stabil (nilai statistiknya tidak berubah terhadap waktu) maupun nonstabil (nilai signal berubah-ubah terhadap waktu). Dengan melakukan pemodelan terhadap signal secara benar, dapat dilakukan simulasi terhadap sumber dan pelatihan sebanyak mungkin melalui proses simulasi tersebut. Sehingga model dapat diterapkan dalam sistem prediksi, sistem pengenalan, maupun sistem identifikasi [1].

Pada penelitian ini digunakan CNN sebagai metode pembelajaran mendalam untuk pengenalan sinyal suara sebagai proses klasifikasi banyak kelas dengan kemampuan pengenalan kata-kata yang terisolasi yang kemudian hasil pengujian final dari CNN digunakan sebagai input basis data yang akan ditanamkan pada mikrokontroller ESP32 menggunakan bantuan tensorflow lite dan microfon jenis MEMS sebagai input suara pada mikrokontroller. Motivasi menggunakan model deep learning CNN adalah untuk menemukan model terbaik yang cocok untuk menyelesaikan proses pengenalan ucapan . Seperti disebutkan sebelumnya, data perekaman memiliki banyak masalah dalam hal pengucapan, noise atau kebisingan pada latar belakang dan alat perekam. Penelitian pada umumnya melakukan ekstraksi ciri pada sinyal suara, namun penelitian ini menggunakan sinyal suara yang diubah ke bentuk spektogram dengan menggunakan algoritma STFT (Short Time Fourier Transform) adalah pengembangan dari FFT (Fast Fourier Transform). Sinyal akan dicuplik selama t detik yang kemudian diterjemahkan dalam domain frekuensi, sehingga sinyal tersebut akan diketahui posisinya dalam domain waktu dan frekuensi, hasil spektrum suara yang berbentuk citra 2D yang disebut spektogram tersebut akan digunakan sebagai input pembelajaran mendalam. Berbagai jenis parameter CNN yang dapat dipelajari telah diuji dan diperbarui untuk menemukan struktur CNN terbaik yang mampu menyelesaikan masalah klasifikasi banyak kelas pada penelitian ini.

2. TINJAUAN PUSTAKA

2.1 Suara

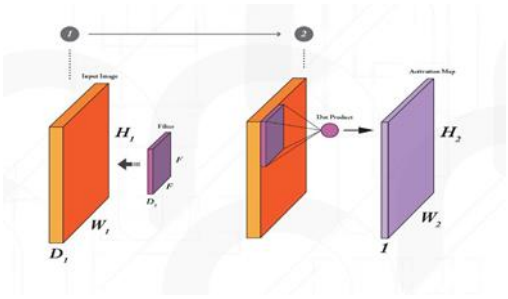
Suara adalah sinyal atau gelombang yang merambat di udara dengan frekuensi dan amplitudo tertentu. Selain melalui udara gelombang suara juga dapat merambat melalui media padat, dan cair. Gelombang suara merambat paling cepat di media padat dan tidak dapat merambat di ruang hampa. Karakteristik gelombang suara ditentukan oleh frekuensi dan amplitudo. Frekuensi adalah jumlah siklus gelombang suara yang terjadi dalam satu detik. Frekuensi akan semakin tinggi apabila getaran gelombang suara semakin cepat. Misalnya apabila seseorang bernyanyi dengan pitch yang tinggi akan memaksa pita suara untuk bergetar lebih cepat sehingga menghasilkan gelombang suara dengan frekuensi yang tinggi. Satuan dari Frekuensi adalah Herzt atau disingkat dengan Hz. Standar frekuensi gelombang suara yang dapat didengar oleh manusia berada diantara 20Hz sampai dengan 20kHz. Frekuensi ini disebut sebagai Frekeunsi Audio. Amplitudo adalah jarak simpangan terjauh dari titik kesetimbangan sebuah gelombang. Satuan amplitudo adalah desibel (dB). Amplitudo menentukan kuat lemahnya suara, semakin

DOI: https://doi.org/10.37277/stch.v33i4.1781

besar amplitudo maka semakin keras suara yang terdengar.

2.2 Convulutional Neural Network

CNN adalah sebuah teknik yang terinspirasi dari cara mamalia — manusia, menghasilkan persepsi visual seperti contoh diatas. Penelitian oleh Hubel dan Wiesel [2] menunjukkan bahwa neuron dalam korteks visual pada otak mamalia tersusun menjadi sebuah *topographical map*, yang setiap level nya fokus pada karakteristik tertentu. Atau dengan kata lain, sebuah gambar direpresentasikan ke dalam level ektraksi fitur (*featural hierarchy)* yang membentuk gambar tersebut secara keseluruhan. Sesuai dengan namanya, CNN memanfaatkan proses konvolusi. Dengan kata lain, bagi sebuah mesin, sebuah gambar memiliki 3 parameter yang perlu diperhatikan, yaitu tinggi (*height*), lebar (*width*), dan tebal/jumlah kanal (*depth*). Proses konvolusi memanfaatkan apa yang disebut sebagai filter. Seperti layaknya gambar, filter memiliki ukuran tinggi, lebar, dan tebal tertentu. Filter ini diinisialisasi dengan nilai tertentu (*random* atau menggunakan teknik tertentu seperti Glorot), dan nilai dari filter inilah yang menjadi parameter yang akan di-*update* dalam proses *learning*. Gambar input CNN selalu berbentuk **kotak**. Proses untuk gambar *non-rectangular* masih belum diketahui [4]. Filter pun mengikuti karakteristik kotak tersebut.



Gambar 2.1 Proses pada Convulutional Layer

Pada setiap posisi gambar, dihasilkan sebuah angka yang merupakan *dot product* antara bagian gambar tersebut dengan filter yang digunakan. Dengan menggeser (*convolve*) filter di setiap kemungkinan posisi filter pada gambar, dihasilkan sebuah *activation map.* Ukuran spasial dari output proses diatas dapat dihitung berdasarkan persamaan 1:

$$H1 = \frac{(N - F + 2P)}{(S + 1)} \ldots\ldots\ldots\ldots (1)$$

dimana N adalah ukuran spasial (tinggi H1 = lebar W1) dari gambar input, F merupakan ukuran spasial filter, P adalah jumlah penambahan angka (umumnya nol) untuk menyesuaikan ukuran gambar, dan S adalah besaran pergeseran filter di setiap proses komputasi. Beberapa hal yang perlu diperhatikan pada convolutional layer:

- Ukuran ketebalan dari sebuah filter selalu mengikuti ketebalan/volume dari gambar input yang digunakan.
- Tinggi (dan lebar) filter (F) pada umumnya berukuran ganjil. Secara intuisi, filter berukuran ganjil memberikan representasi yang lebih baik karena mencakup bagian kiri dan kanan yang "seimbang".
- Dalam sebuah convolutional layer, filter-filter yang digunakan berukuran sama, untuk kemudahan proses komputasi.
- Jumlah filter (K) yang digunakan dalam sebuah convolutional layer adalah kelipatan 2 (powers of 2). Beberapa library memiliki subroutine khusus untuk komputasi kernel/dimensi berkelipatan 2 yang dapat meningkatkan efisiensi.
- Besaran zero padding (P) umumnya menyesuaikan agar ukuran spasial dari output yang dihasilkan tetap sama dengan ukuran spasial input.

Secara keseluruhan, bila input sebuah convolutional layer adalah gambar dengan ukuran W1 x H1 x D1, output dari layer tersebut adalah sebuah "gambar" baru dengan ukuran W2 x H2 x D2, dimana:

$$\boldsymbol{W2 = \frac{(W1 - F + 2P)}{(S+1)}, D2 = K} \ldots\ldots\ldots\ldots \boldsymbol{(2)}$$

dimana:

- K adalah jumlah filter yang digunakan.
- F adalah ukuran spasial dari filter (lebar/tinggi).
- S adalah stride, atau besar pergeseran filter dalam konvolusi.
- P adalah padding, jumlah penambahan nol pada gambar.

2.3 Short-Time Fourier Transform (STFT)

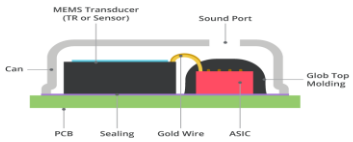
Algoritma Short Time Fourier Transform(STFT) adalah pengembangan dari Transformasi Fourier. Transformasi Fourier hanya dapat menghasilkan informasi apakah

DOI: https://doi.org/10.37277/stch.v33i4.1781

suatu sinyal memiliki komponen frekuensi tertentu atau tidak. Hasil Transformasi Fourier tidak dapat menunjukkan kapan komponen frekuensi tersebut terjadi. Sehingga Transformasi Fourier hanya cocok diterapkan pada sinyal stasioner yaitu sinyal yang komponen frekuensinya tidak berubah seiring dengan perubahan waktu atau semua komponen frekuensi terjadi di semua waktu. Suara adalah sinyal non-stationary atau sinyal yang komponen frekuensinya berubah terhadap waktu. Untuk mengetahui kapan suatu komponen frekuensi dari sinyal suara terjadi, dapat digunakan algoritma STFT.

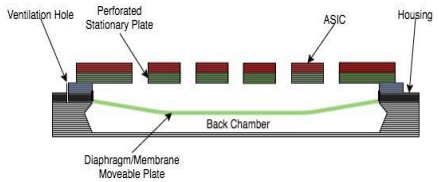
2.4 Mikrofon MEMS

Mikrofon MEMS adalah transduser elektro-akustik yang menampung sensor (MEMS) dan sirkuit terintegrasi khusus aplikasi (ASIC) dalam satu paket. Sensor mengubah tekanan suara masuk variabel menjadi variasi kapasitansi yang diubah ASIC menjadi output analog atau digital.



Gambar 2.2 Penampang mikrofon MEMS

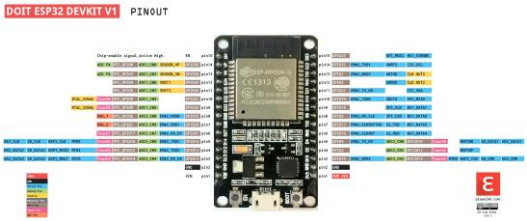
Seperti kebanyakan teknologi MEMS, mikrofon MEMS dibuat menggunakan wafer silikon semikonduktor dan proses yang sangat otomatis.. Mikrofon dengan teknologi ASIC dirancang agar sesuai dengan elemen transduser mikrofon MEMS. Ini menggunakan pompa muatan untuk menempatkan muatan tetap antara pelat stasioner dan membran mikrofon.



Gambar 2.3 Prinsip Kerja Mikrofon MEMS

2.5 ESP32

ESP32 adalah mikrokontroler yang dikenalkan oleh Espressif System merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi Internet of Things. terlihat pada gambar di atas merupakan pin out dari ESP32. Pin tersebut dapat dijadikan input atau output untuk menyalakan LCD, lampu, bahkan untuk menggerakan motor DC.

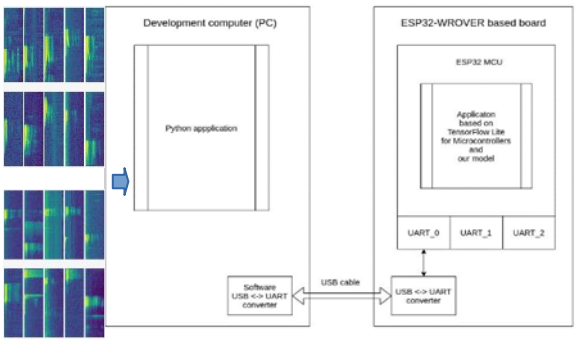


Gambar 2.4 Breakout ESP32 Devkit

ESP32 ini memiliki tegangan operasi **3.3V** berbeda dengan mikrokontroler jenis AVR , sehingga untuk membuat suatu rangkaian elektronik menggunakan ESP32 harus di perhatikan bahwa supply listrik pada rangkaian *tidak boleh lebih dari 3.3V* .

3. METODE PENELITIAN

3.1 Diagram Blok



3.1 Diagram Blok Sistem

Perancangan dan pembuatan perangkat pengenalan suara menggunakan CNN sebagai metode pembelajaran mendalam untuk pengenalan sinyal suara sebagai proses klasifikasi banyak kelas dengan kemampuan pengenalan kata-kata yang terisolasi yang kemudian hasil pengujian final dari CNN digunakan sebagai input basis data yang akan ditanamkan pada mikrokontroller ESP32 menggunakan bantuan tensorflow lite dan microfon jenis MEMS sebagai input suara pada mikrokontroller. Motivasi menggunakan model deep learning CNN adalah untuk menemukan

DOI: https://doi.org/10.37277/stch.v33i4.1781

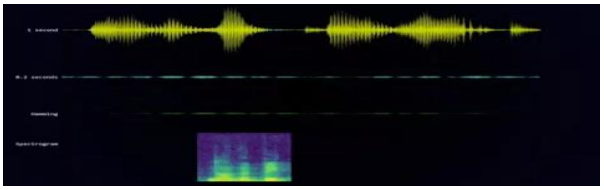
model terbaik yang cocok untuk menyelesaikan proses pengenalan ucapan . Seperti disebutkan sebelumnya, data perekaman memiliki banyak masalah dalam hal pengucapan, noise atau kebisingan pada latar belakang dan alat perekam. Penelitian pada umumnya melakukan ekstraksi ciri pada sinyal suara, namun penelitian ini menggunakan sinyal suara yang diubah ke bentuk spektogram dengan menggunakan algoritma STFT (Short Time Fourier Transform) adalah pengembangan dari FFT (Fast Fourier Transform). Sinyal akan dicuplik selama waktu t detik dan kemudian diterjemahkan dalam domain frekuensi, sehingga sinyal tersebut akan diketahui posisinya dalam domain waktu dan frekuensi, hasil spektrum suara yang berbentuk citra 2D yang disebut spektogram tersebut akan digunakan sebagai input pembelajaran mendalam. Berbagai jenis parameter CNN yang dapat dipelajari telah diuji dan diperbarui untuk menemukan struktur CNN terbaik yang mampu menyelesaikan masalah klasifikasi banyak kelas pada penelitian ini.

3.2 Penyiapan Data Pelatihan

Pada penelitian ini digunakan Speech Commands Dataset, dataset ini berisi rekaman audio 100.000 kata perintah dengan lama durasi 1 detik dengan 20 jenis kata dalam bahasa inggris seperti up, down ,right,left dan beberapa tambahan perintah bahasa indonesia seperti buka dan tutup Untuk menambah kumpulan data pelatihan , dilakukan perekaman kebisingan latar sekitar, seperti kebisingan di kantor, rumah dan acara TV untuk memberikan sejumlah besar data acak.

3.3 Pengambilam Fitur Spektogram

Dengan data pelatihan yang telah didapat, langkah selanjutnya adalah menentukan fitur yang akan dilatih untuk jaringan saraf tiruan , dalam hal ini tidak mungkin untuk memasukkan gelombang audio dalam bentuk mentah (raw) sebagai input ke jaringan saraf . Dengan cara seperti itu tidak mungkin memberikan hasil output yang baik. Pendekatan metode yang digunakan untuk pengenalan kata dalam penelitian ini adalah melakukan konversi pengenalan bentuk input gelombang menjadi pengenalan citra. Untuk mendapatkan spektogram dari sampel audio, sampel dipecah menjadi bagian-bagian kecil dan kemudian dilakukan transformasi Fourier diskrit pada setiap bagian. Dengan itu akan diperoleh frekuensi yang ada dalam potongan audio dengan menempatkan irisan frekuensi ini secara bersama-sama akan diperoleh spektogram sampel.



Gambar 3.2 Pengambilan Sampel Spektogram

3.4 Pelatihan Model

Selanjutnya adalah melakukan pelatihan pada model yang telah dibuat pada sebelumnya model terdiri dari data pelatihan, pengujian, dan validasi yang dibuat pada langkah sebelumnya. Kemudian model dimasukkan ke dalam set data TensorFlow untuk selanjutya dilakukan proses berulang, pengacakan data , dan hasil output kemudian masuk ke dalam variabel batch. Setelah melakukan beberapa penyesuaian parameter dan dengan beberapa arsitektur model yang berbeda dan maka diperoleh waktu untuk melatih, akurasi dan ukuran model yang terbaik . Dengan menggunakan konvulsi 2 dimensi dan beberapa lapisan konvolusi, diikuti oleh lapisan max-pooling, selanjutnya lapisan konvolusi lain dan lapisan max-pooling. Hasil dari ini dimasukkan ke dalam lapisan yang terhubung secara padat dan akhirnya menuju neuron keluaran Kemudian hasil output dapat dilihat pada tabel 3.1, dengan akurasi sebagai berikut:

Tabel 3.1 Hasil data Pelatihan pada Suara

Dataset	Accuracy
Training Dataset	0.9683
Validation Dataset	0.9567
Test Dataset	0.9562

3.5 Perancangan Firmware

Setelah mendapatkan bentuk model yang terlatih , hasil tersebut perlu dikonversi untuk digunakan di TensorFlow Lite. Proses konversi

DOI: https://doi.org/10.37277/stch.v33i4.1781

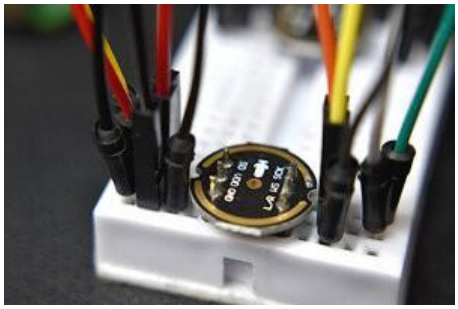
ini dilakukan dengan model hasil dari pengenalan suara dalam format fie ekstensi .h dan mengubahnya menjadi versi yang jauh lebih ringkas yang dapat dijalankan di perangkat ESP32. Untuk memeperoleh file yang akan ditanam di mkrokontroler dapat dijalankan baris perintah untuk menghasilkan kode C yang dapat kita kompilasi ke dalam mikrokontroler.

4. HASIL DAN PEMBAHASAN

Tujuan dari melakukan pengujian pada sistem yaitu untuk mengetahui kinerja dari perangkat , dan kemampuan secara keseluruhan. Data hasil pengujian dapat menampilkan kekurangan dan kelebihan dari sistem.

4.1 Pengujian Microfon INMP441 Mode Input Mono

Uji fungsi antara mikrofon MEMS INMP441 dengan komunikasi I2S pada ESP32 adalah dengan memberikan sinyal akustik pada mikrofon. Pengujian ini dilakukan dengan menggunakan frekuensi yang dihasilkan oleh suara manusia atau generator fungsi audio pda pengujian ini digunakan frekensi sebesar 3114 KHz. Data audio kemudian dapat dianalisis menggunakan Fast Fourier Transform (FFT), di mana frekuensi input dapat diverifikasi dalam spektrum frekuensi yang direkam oleh mikrofon INMP441.

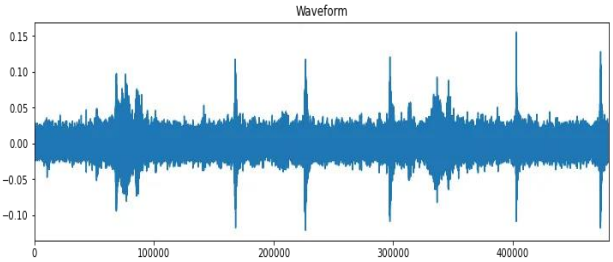


Gambar 4.1 Pengujian mikrofon MEMS INMP441 input mono

Proses lengkapnya adalah merekam 1 detik kebisingan latar belakang / background (pada laju sampel 44,1 kHz) kemudian rekam selama 5 detik. Hasil rekaman dapat disimpan dalam bentuk file .wav dan hapus kebisingan latar belakang dari data 5 detik yang diambil kemudian pilih frekuensi puncak dari data respon frekuensi dan tampilkan dalam bentuk plot grafik deret waktu.

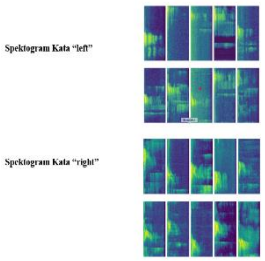
4.2 Hasil Pengujian Input Gelombang Suara Menjadi Spektogram

File audio yang diambil memiliki format ".wav" dan disebut sebagai bentuk gelombang. Bentuk gelombang merupakan deret waktu dengan amplitudo sinyal pada setiap waktu tertentu. Jika salah satu sampel bentuk gelombang tersebut divisualisasikan, maka akan diperoleh Gambar 4.2 sebagai berikut



Gambar 4.2 bentuk normalisasi Sinyal Suara

selanjutnya adalah mengonversi file bentuk gelombang menjadi spektrogram.fungsi ini terdapat di Tensorflow yaitu fungsi tf.signal.stft yang menerapkan short-time Fourier transform (STFT) untuk mengubah audio menjadi domain waktu-frekuensi. Operator tf.abs kemudian diterapkan untuk menghapus fase sinyal dan hanya mempertahankan besarnya. Perlu diperhatikan bahwa ada beberapa parameter seperti frame_length dan frame_step yang akan mempengaruhi hasil sepktogram yang dihasilkan



Gambar 4.3 Hasil Pengujian Spektogram

4.3 Hasil Pengujian Sistem Minimum ESP 32

Pengujian tegangan output adalah salah satu cara untuk memastikan bahwa ESP 32 dapat menghasilkan tegangan yang stabil dan sesuai dengan spesifikasi. Pengujian ini dilakukan dengan menggunakan multimeter dan mengukur tegangan yang dihasilkan oleh ESP 32 pada titik tertentu. Hasil pengujian harus sesuai dengan spesifikasi tegangan

DOI: https://doi.org/10.37277/stch.v33i4.1781

output ESP 32 yang ditentukan sebelumnya. Pengujian timer juga dilakukan untuk memastikan bahwa ESP 32 dapat menjalankan timer dengan akurasi yang baik. Pengujian ini dilakukan dengan memprogram ESP 32 untuk mengukur interval waktu tertentu. dan membandingkannya dengan interval yang sebenarnya. Hasil pengujian harus sesuai dengan spesifikasi timer ESP 32 yang ditentukan sebelumnya. Tabel berikut menunjukkan hasil pengujian tegangan output dan timer pada sistem minimum ESP 32:

Tabel 4.1 Pengujian Tegangan Output dan timer pada ESP32

Pengujian	Spesifikasi	Hasil	Keterangan
Tegangan Output	3.3V	3.29V	Sesuai spesifikasi
Timer	1 detik	0.98 detik	Sesuai spesifikasi

Dari hasil pengujian, dapat diketahui bahwa ESP 32 dapat menghasilkan tegangan output yang stabil dan menjalankan timer dengan akurasi yang baik, sesuai dengan spesifikasi.

4.4 Hasil Pengujian Motor Servo

Pengujian putaran motor servo adalah salah satu cara untuk memastikan bahwa ESP 32 dapat memutar motor servo dengan akurasi yang baik. Pengujian ini dilakukan dengan memprogram ESP 32 untuk memutar motor servo pada sudut yang ditentukan dan mengukur putaran yang dihasilkan.

Hasil pengujian harus sesuai dengan spesifikasi putaran motor servo yang ditentukan sebelumnya. Untuk memastikan hasil yang akurat, pengujian putaran motor servo dilakukan sebanyak 6 kali. Setiap pengujian dilakukan pada sudut yang berbeda dan hasil pengujian dicatat dan dianalisis

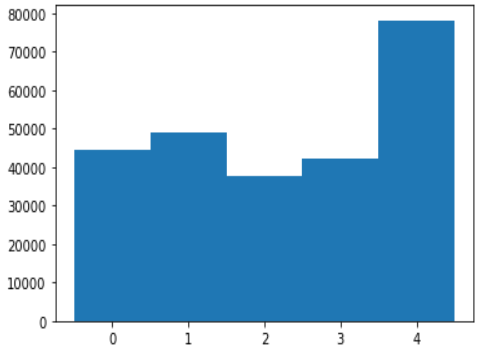
Tabel 4.2 Pengujian Motor Servo

Pengujian Ke	Sudut	Putaran	Deviasi	Keterangan
1	30 derajat	29.5 derajat	-0.5 derajat	Sesuai spesifikasi
2	60 derajat	59.7 derajat	-0.3 derajat	Sesuai spesifikasi
3	90 derajat	89.8 derajat	-0.2 derajat	Sesuai spesifikasi
4	120 derajat	120.2 derajat	0.2 derajat	Sesuai spesifikasi
5	150 derajat	149.5 derajat	-0.5 derajat	Sesuai spesifikasi
6	180 derajat	178.6 derajat	-1.4 derajat	Sesuai spesifikasi

Dari hasil pengujian, dapat diketahui bahwa ESP 32 dapat memutar motor servo dengan akurasi yang baik, sesuai dengan spesifikasi. Deviasi yang terjadi pada setiap pengujian tidak melebihi batas toleransi yang ditentukan.

4.5 Hasil Proses Pelatihan dan Analisa CNN

Pada proses training dapat dilihat hasil pengelompokan distribusi dari 5 buah kata yang akan diproses dengan rata-rata masing masing contoh kata adalah sebesar 50.000 sampel yang terdiri dari kata {'forward': 44520, 'backward': 49000, 'left': 37520, 'right': 42220, '_invalid': 78183} hasil ini kemudian dapat kita lihat dalam bentuk histogram seperti Gambar 4.6 yang ada dibawah ini

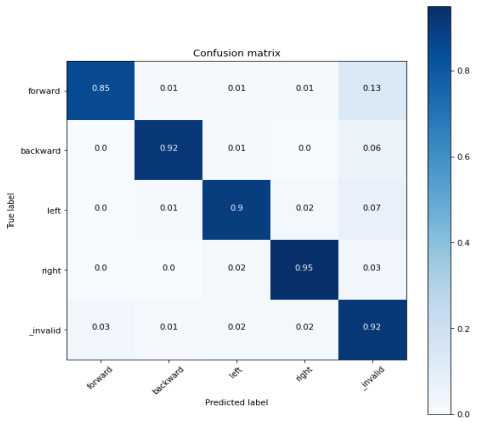


Gambar 4.6 Plot Histogram dari variabel input

Pada hasil running ini pengulangan atau epoch dilakukan sebanyak 10 kali dengan waktu rata-rata 47 detik per step, proses ini dapat dikatakan cukup cepat karena menggunakan bantuan GPU nvidia 2060 untuk melakukan pelatihan . Pada pelatihan ini diperoleh akurasi rata-rata sebesar

DOI: https://doi.org/10.37277/stch.v33i4.1781

92 %. Untuk melihat seberapa baik hasil dari pelatihan maka data diatas dapat kita ubah dengan fungsi plot_confussion_matrix sehingga dapat disajikan sebagai berikut

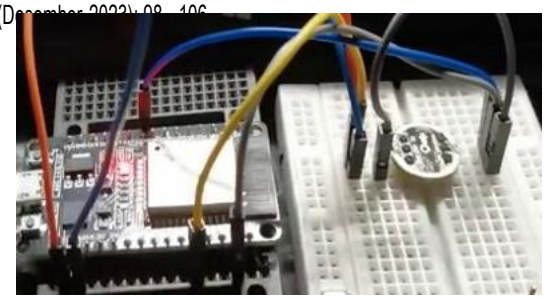


Gambar 4.7 Confussion Matrix dari Pelatihan

Pada grafik confussion matriks diatas dapat dijabarkan bahwa pelatihan terhadap 5 variabel kata berjalan dengan baik sebagai contoh pada kata backward, akurasi antara predicted label dengan true label adalah sebesar 0.92 atau 92 % ,begitu juga dengan 4 variabel lainnya yang menunjukan kecocokan dengan nilai akurasi diatas 90 %, ini menunjukan proses pelatihan berjalan dengan baik.

4.6 Hasil Pengujian Akhir Sistem

Dari hasil uji akhir diatas dapat dijabarkan bahwa pelngujian berjalan dengan akurasi antara predicted label dengan true label adalah sebesar 0.92 atau 95 % sampai dengan 97% ,begitu juga dengan 4 variabel lainnya yang menunjukan kecocokan dengan nilai akurasi diatas 90 %, ini menunjukan proses pengujian (Testing model) memberikan hasil yang baik untuk melakukan pengenalan kata dengan noise yang cukup beragam, untuk noise pengujian dilakukan dengan menguji variabel yang berisi invalid. Setelah melakukan pengujian ini selanjutnya adalah melakukan pengujian pada perangkat esp32.



Gambar 4.8 Pengujian Keseluruhan Sistem

Pada pengujian Alat diatas dapat diketahui pengujian berhasil dengan melihat di bagian terminal pada VS code terdeteksi ucapan “forward “ dengan output dari terminal yaitu Detected command forward, begitu juga dengan perintah lainnya dengan waktu deteksi rata-rata sebesar 103 ms, untuk lebih jelasnya dapat dilihat pada video yang terdapat pada lampiran. Dengan demikian pengujian sistem pengenalan kata dari perangkat telah berhasil.

Tabel 4.3 Hasil Pengujian keseluruhan

Perintah Suara	Hasil	Keterangan
Left	Putaran motor servo ke kiri	Sesuai perintah
Right	Putaran motor servo ke kanan	Sesuai perintah
Forward	Putaran motor servo ke depan	Sesuai perintah

Dalam pengujian ini, sistem diuji dengan 3 perintah suara yaitu left, right, forward. Hasil pengujian menunjukkan bahwa sistem dapat menerima dan menafsirkan perintah suara dengan benar dan mengendalikan putaran motor servo sesuai dengan perintah yang diberikan.

V. KESIMPULAN

Berdasarkan hasil pengujian dan Analisa pada perangkat pengenalan kata dan suara dapat disimpulkan bahwa :

1. Tingkat akurasi pada tiap kata yang dikenali berdasarkan grafik convusion matrix memiliki nilai diatas 90% dengan jumlah sampel sebesar 50.000
2. Suara Latar (noise) tidak mempengaruhi proses proses pengenalan kata dan suara, ini menunjukan data noise yang dilatih berhasil

DOI: https://doi.org/10.37277/stch.v33i4.1781

3. Kemampuan mikrofon MEMS mampu untuk mendeteksi frekuensi tertentu di sekitar frekuensi input.

VI. SARAN

Untuk pengembangan selanjutnya perangkat pengenalan suara dapat digunakan untuk mengontrol perangkat rumah hingga peralatan terapi medis

VII DAFTAR PUSTAKA

A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 6645-6649.

C. S. Chan, C. C. Chang, and C. Y. Lee, "Speech Recognition Using Convolutional Neural Network with Long Short-Term Memory," in 2019 IEEE 9th International Conference on Advanced Computational Intelligence (ICACI), 2019, pp. 114-117.

G. Hinton, S. Osindero, and Y. W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," Neural Comput., vol. 18, no. 7, pp. 1527-1554, 2006.

J. Li, M. Hou, and Z. Zhang, "A Convolutional Neural Network Based Speech Recognition System for Smart Home," IEEE Access, vol. 7, pp. 113941-113950, 2019.

M. F. A. Rasheed, M. S. M. Aras, and M. A. M. Basri, "Voice Command Recognition System Using Convolutional Neural Network for Smart Home Application," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 6, pp. 321-328, 2019.

O. Vinyals, A. Graves, and N. Jaitly, "A Neural Conversational Model," arXiv preprint arXiv:1506.05869, 2015.

S. A. Wulandari, A. R. Pratama, and D. S. Wijaya, "Sistem Pengenalan Perintah Suara Berbasis Android dengan Algoritma CNN," J. Teknol. dan Sist. Inf., vol. 6, no. 2, pp. 141-148, 2021.

S. O. Arik, M. K. Akgul, and M. E. Eren, "Voice Command Recognition System Using Convolutional Neural Networks," in 2017 25th Signal Processing and Communications Applications Conference (SIU), 2017, pp. 1-4.

Y. H. Chen, Y. T. Chen, and C. C. Huang, "A Real-Time Voice Command Recognition System Based on Convolutional Neural Network," Sensors, vol. 20, no. 11, p. 3261, 2020.

Y. Zhang, N. Togneri, and B. Benatallah, "A Convolutional Neural Network Based Approach for Speech Recognition," in 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018, pp. 3608-3613.