

Penerapan CNN dan CNN pada Aplikasi Mobile Pendeteksi Kematangan Buah Semangka Berbasis *Audio*

1st Gisky Rahmada Putri
Prodi S1 Teknik Telekomunikasi
Universitas Telkom
Bandung
Bandung, Jawa Barat
giskyrhadmap@student.telkomuniversity.ac.id

2nd Gelar Budiman
Prodi S1 Teknik Telekomunikasi
Universitas Telkom
Bandung
Bandung, Jawa Barat
gelarbudiman@telkomuniversity.ac.id

3rd Yulinda Eliskar
Prodi S1 Teknik Telekomunikasi
Universitas Telkom
Bandung
Bandung, Jawa Barat
yulindaeliskar@telkomuniversity.ac.id

Abstrak - Semangka banyak diminati masyarakat karena kandungan pada buahnya yang memiliki banyak air sehingga sangat segar saat dikonsumsi, selain itu banyak juga segudang manfaat didalamnya. Akan tetapi keminatan terhadap buah semangka ini tidak diikuti dengan pengetahuan masyarakat terhadap matang atau tidaknya buah semangka yang dijual di pasaran. Oleh karena hal tersebut menjelaskan bahwa membutuhkan alat pendeteksi untuk membantu masyarakat saat pemilihan buah semangka. Dengan alasan ini lah peneliti ingin menciptakan alat pendeteksi semangka berbasis android menggunakan machine learning serta deep learning. machine learning disini menggunakan metode CNN serta MFCC, lalu pada deep learning menggunakan metode CNN. Adanya dua alat tersebut dilakukan perbandingan akurasi pada semangka yang akan diuji. Dengan menepuk buah semangka dapat menghasilkan output yang berbeda dari nyaringnya, sehingga buah semangka yang diketuk dapat diketahui tingkat kematangannya. Bunyi nyaring yang dikeluarkan pada buah semangka dihasilkan karena kandungan yang terdapat pada buah semangka sendiri terutama pada kandungan airnya. Pada buah semangka yang sudah matang tentu saja kandungan air sudah banyak sehingga bunyi yang dihasilkan tidak terlalu nyaring, sedangkan buah muda akan terdengar nyaring karena masih sedikit pada kandungan airnya. Dari hasil penelitian yang sudah diuji menghasilkan tingkat akurasi pada CNN yaitu 70 hingga 80%. Dengan demikian penelitian ini memiliki tujuan terutama untuk mengedukasi masyarakat terhadap kematangan buah semangka, serta menghindari penipuan saat membelinya.

Kata kunci — Semangka, Matang, Machine learning, Deep learning, Android.

I. PENDAHULUAN

Semangka merupakan buah yang cukup populer di Indonesia. Semangka termasuk dalam keluarga labu labuan [1]. Kemiripan buah semangka dengan buah kelapa dalam hal pendeteksian kematangan itu sama, dengan mengetuk buahnya dapat diketahui seberapa tingkat kematangannya [2]. Buah semangka yang diketuk dapat diketahui tingkat kematangannya dikarenakan adanya kandungan air di dalam buah, kematangan dari buah dapat dibedakan dari nyaringnya. Namun untuk kondisi pada pedagang, tentu saja bunyi tersebut hanya bersifat subjektif karena keadaan sekitar yang berbeda. Sehingga inovasi teknologi dengan memanfaatkan *microphone handphone* untuk pendeteksian buah semangka bagi penjual dan terutama bagi pembeli diperlukan.

Untuk mendapatkan hasil yang relevan, tidak memungkinkan untuk membeli semangka di tempat yang sepi, sehingga diperlukan metode CNN (Convolutional Neural Network), yang beroperasi menggunakan tiga lapisan: lapisan konvolusional, lapisan penyatuan, dan lapisan yang terhubung penuh. Setelah melakukan percobaan, metode dengan akurasi tertinggi akan dipilih dan kemudian diintegrasikan ke dalam aplikasi mobile berbasis Android. Perancangan aplikasi mobile menggunakan Flutter diawali dengan proses persiapan Firebase, dilanjutkan dengan pembuatan kode sesuai dengan kebutuhan, meliputi, tombol-tombol untuk merekam *audio*, mengunggah *audio* ke Firebase, proses menunggu pendeteksian, hingga tampilan output hasil klasifikasi.

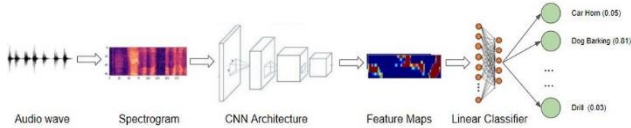
II. KAJIAN TEORI

Deep learning adalah subbidang pembelajaran mesin yang berfokus pada pengembangan algoritma yang terinspirasi oleh struktur dan fungsi jaringan saraf otak manusia, yang dikenal sebagai jaringan saraf tiruan. *Deep learning* memiliki kemampuan untuk memproses dan memahami data yang sangat kompleks, seperti gambar, suara, dan teks, dan telah menghasilkan kemajuan yang signifikan dalam aplikasi seperti pengenalan gambar, pemrosesan bahasa alami, pengenalan suara, dan banyak lagi.

Convolutional Neural Network (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada data *image*. CNN bisa digunakan untuk mendeteksi dan mengenali objek pada sebuah *image* dan *audio*. Secara garis besar CNN tidak jauh beda dengan *neural network* biasanya. CNN terdiri dari Neuron yang memiliki *weight*, bias dan *activation function*. Klasifikasi *audio* ini melibatkan pembelajaran untuk mengklasifikasikan suara dan memprediksi kategori suara itu. Jenis soal ini dapat diterapkan pada banyak skenario praktis, misalnya untuk mendeteksi kematangan pada buah semangka. Kita akan mulai dengan file suara, mengonversinya menjadi spektrogram, memasukkannya ke dalam model CNN, dan menghasilkan prediksi tentang kelas tempat suara tersebut berada. Kumpulan data yang cocok untuk jenis suara yang berbeda. Kumpulan data ini berisi sampel *audio*, bersama dengan label kelas untuk setiap

sampel yang mengidentifikasi jenis suaranya, berdasarkan masalah yang ingin diatasi.

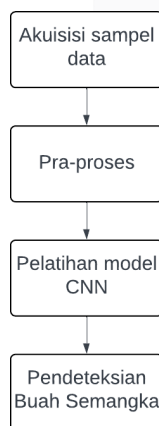
Convolutional layer merupakan komponen yang penting pada CNN. Konvolusi merupakan sebuah istilah yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Tujuan dari *layer* ini yaitu untuk mengekstraksi fitur



Gambar 2.1 Cara Kerja CNN

dari suatu citra *input*. Konvolusi akan menghasilkan transformasi linier dari data input sesuai dengan informasi spasial pada data.

Selain itu, pada CNN dibutuhkan *Learning rate* adalah salah satu parameter yang berperan penting dalam proses dan menjadi penentu banyaknya *weight* yang akan diperbaharui dalam proses *training*. Nilai *Learning rate* berkisar antara 0 sampai 1. *Learning rate* juga menentukan seberapa cepat model menyesuaikan dengan masalah. Semakin besar nilai *learning rate* maka proses *training* semakin cepat. Sedangkan nilai *learning rate* yang terlalu kecil menyebabkan proses *stuck*. Oleh karena itu dibutuhkan percobaan untuk mencari nilai *learning rate* yang optimal.



Gambar 2.2 Diagram CNN

Gambar diatas merupakan diagram yang menunjukkan tahapan proses yang akan dilakukan untuk mendapatkan model latih yang digunakan dalam mendeteksi kematangan pada buah semangka. Tahap awal dimulai dengan melakukan akuisisi dataset berupa mengambil data *audio* pada buah semangka yang ditepuk dan terbagi ke dalam tiga kelas, yaitu belum matang, matang, matang sekali. Selanjutnya, tahap kedua data *audio* yang sudah didapatkan diolah dalam tahap pre-processing dengan cara mengkonversi data *audio* ke dalam bentuk WAV, kemudian tahap ketiga adalah proses pelatihan dataset menggunakan *script algoritma deep learning* dengan metode CNN. Proses pelatihan dataset akan menghasilkan model *deep learning* yang dapat mengenali suara pada buah semangka.

A. Arsitektur *Convolutional Neural Network*

Arsitektur *Convolutional Neural Network* (CNN) adalah jenis jaringan saraf tiruan jaringan saraf tiruan yang dirancang khusus untuk memproses data yang berbentuk kisi-kisi. Berikut ini adalah komponen utama arsitektur CNN:

- Lapisan Konvolusional (*Convolutional Layer*): Lapisan ini menggunakan filter untuk melakukan konvolusi pada input. Filter-filter ini membantu mendeteksi fitur seperti tepi, sudut, dan tekstur pada gambar.
- Fungsi Aktivasi: Setelah lapisan konvolusi, fungsi aktivasi seperti *Rectified Linear Unit* (ReLU) sering digunakan untuk memperkenalkan non-linearitas ke jaringan.
- Lapisan Penyatuan (*Pooling Layer*): Lapisan ini mengurangi dimensi *spasial* dari peta fitur yang dihasilkan oleh lapisan konvolusi. Penyatuan maksimum atau penyatuan rata-rata biasanya digunakan.
- Lapisan Normalisasi *Batch*: Lapisan ini membantu mengurangi pergeseran kovariat internal dan mempercepat pelatihan dengan menormalkan input dari lapisan sebelumnya.
- Lapisan Terhubung Penuh (FC): Lapisan ini menggabungkan informasi dari lapisan sebelumnya dan menghasilkan keluaran. Pada akhirnya, lapisan ini dapat berfungsi sebagai klasifikasi atau regresi lapisan *output*, tergantung pada tugasnya.
- Lapisan *Dropout*: Lapisan ini dapat ditambahkan untuk mengurangi *overfitting* dengan menonaktifkan beberapa unit secara acak menonaktifkan beberapa unit secara acak selama pelatihan.
- Fungsi *Softmax*: Digunakan pada lapisan *output* untuk menghasilkan distribusi probabilitas untuk setiap *class* dalam tugas klasifikasi.
- Fungsi Kerugian (*Loss Function*): Fungsi kerugian digunakan untuk mengukur seberapa jauh prediksi model dari kebenaran selama pelatihan jaringan. Ini digunakan untuk menyesuaikan model parameter selama proses pembelajaran.

Beberapa arsitektur CNN yang terkenal termasuk LeNet, AlexNet, VGGNet, GoogLeNet (Inception), ResNet, dan lainnya. Setiap arsitektur memiliki keunikan struktur dan biasanya dikembangkan untuk menyelesaikan tugas-tugas tertentu seperti gambar klasifikasi, deteksi objek, atau segmentasi gambar.

III. METODE

A. *Deep Learning* CNN

Dataset sudah terbentuk dilanjutkan dengan proses klasifikasi menggunakan CNN yaitu dimulai dengan pembagian data menggunakan *test_size*, lalu setelah itu proses dilakukan sesuai dengan metode CNN yaitu menggunakan jarak *Euclidean*, setelah itu ditentukan parameter nilai K yang akan digunakan serta dilihat akurasi yang baik agar tahap selanjutnya yaitu prediksi mendapatkan hasil terbaik. Setelah semua selesai prediksi dilakukan secara berulang agar dapat dilakukan evaluasi terhadap rancangan yang digunakan.

Pendeteksian kematangan buah semangka *Colabs* dilakukan dengan memperhatikan ke efisiensi *coding*, lalu akurasi, *precision*, waktu, *recall*, *F1_score*. Setelah diperoleh hasil pendeteksian lalu sistem pun selesai hanya tinggal membuat aplikasinya.

Meskipun semuanya berkaitan dengan kinerja model klasifikasi, masing-masing matriks memberikan informasi yang berbeda. Empat metrik evaluasi yang digunakan untuk mengukur kinerja model adalah akurasi, presisi (*precision*), *recall*, dan skor f1 diberikan di sini:

- Akurasi adalah ukuran seberapa baik model klasifikasi mengklasifikasikan data secara keseluruhan. Ini adalah rasio antara jumlah prediksi yang benar benar positif dan benar negatif dengan jumlah data yang dievaluasi secara keseluruhan. Akurasi memberikan gambaran tentang bagaimana model bekerja secara keseluruhan.
- Presisi menunjukkan seberapa baik model berhasil menghindari kesalahan dengan memprediksi positif untuk sampel tertentu. Ini dihitung sebagai rasio antara prediksi positif yang benar, atau prediksi positif yang benar, dan total prediksi positif.
- Sejauh mana model dapat menemukan atau mengidentifikasi kelas positif dengan benar ditentukan oleh *recall*. Ini adalah rasio antara total jumlah data yang sebenarnya positif (*true positive* dan *false negative*). *Recall* menunjukkan seberapa baik model mengidentifikasi semua sampel yang sebenarnya positif.
- F1-score* menggabungkan informasi tentang presisi dan *recall* dan berguna ketika terjadi *trade-off* antara keduanya. Ini adalah rata-rata harmonis dari presisi dan *recall*, yang memberikan pengukuran keseluruhan yang seimbang tentang kinerja model.

B. Hiperparameter

Hyperparameter adalah percobaan dalam proses pengujian untuk memilih *hyperparameter* yang optimal untuk kinerja model. Metode sistem ini menggunakan pengujian *hyperparameter* yaitu *optimizer*, *learning rate*, *epoch*, dan *batch size* dengan membandingkan performa model yang terbaik. Pengujian *hyperparameter* yang memiliki performa model terbaik akan dievaluasi dengan menggunakan *confusion matrix*. *Confusion matrix* merupakan alat ukur standar yang digunakan untuk mengetahui seberapa akurat hasil estimasi dari suatu sistem klasifikasi. *Confusion matrix* dibentuk dari empat buah informasi, yaitu *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, dan *False Negative (FN)*. Berikut ini adalah penjelasan dari keempat hasil klasifikasi *Confusion Matrix* simulasi yang akan dilakukan pada penelitian ini.

Dari matriks kebingungan, matriks pengukuran dapat dibuat untuk mendapatkan nilai akurasi, presisi, *recall*, dan *f1-score*.

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

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

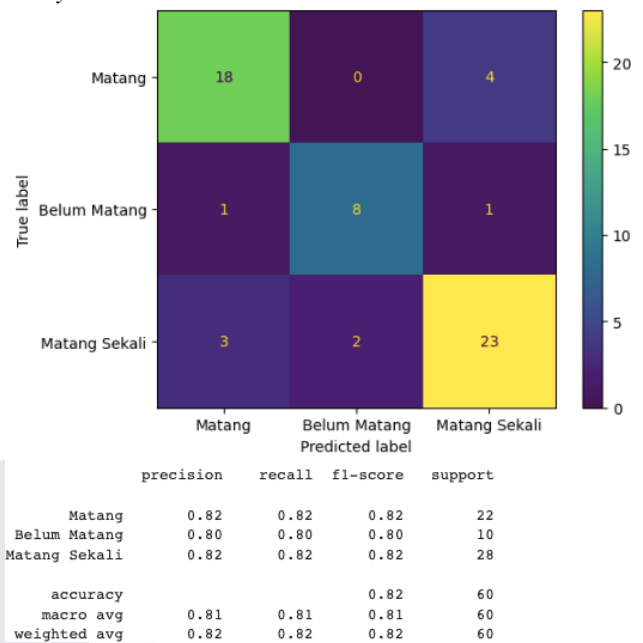
$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = 2 \frac{Recall \times Precision}{Recall + Precision} \quad (4)$$

C. Training dan testing

Proses *training* dimulai dari pengambilan data *audio* dengan cara menepuk bagian permukaan semangka, kemudian pada tahap pengolahan data *audio* akan dikonversi kedalam bentuk WAV, lalu data yang sudah di konversi dilakukan pelatihan data *audio* yang terbagi menjadi 3 kelas yaitu belum matang, matang, matang sekali. Hasil dari proses pelatihan berupa model yang dilatih. Kemudian disimpan dalam *Google Drive* karena akan menggunakan *Google Colabs*.

Sedangkan proses *testing* dimulai dengan file *audio* yang akan diuji dimasukkan kedalam *script coding* dan diproses menggunakan model training yang terlatih atau model akhir. Proses testing ini akan menghasilkan keluaran akurasi dan prediksi belum matang, matang, atau matang sekali. Implementasi pada aplikasi ini adalah dengan tujuan menghasilkan keluaran berupa model yang akurat dengan algoritma CNN. Untuk mendapatkan keluaran tersebut, algoritma CNN memerlukan input berupa dataset *audio* yang sudah di konversi ke wav yang dibagi menjadi 3 kelas yaitu belum matang, matang, matang sekali, setelah sampel data siap, selanjutnya melakukan proses *import library* dan *install library*.



Gambar 3.1 Hasil Pendeteksian pada Python

Selanjutnya pada gambar di atas adalah salah satu contoh hasil pendeteksi sampel dari 5 kali percobaan yang sudah tersedia mendapatkan hasil *accuracy* 82%, *precision*, *recall*, dan *F1-score* 81%. Hasil dari pelatihan ini menghasilkan model *deep learning* yang dapat digunakan dalam mendeteksi kematangan pada buah semangka baru yang akan dideteksi.

IV. HASIL DAN PEMBAHASAN

Pada metode CNN ini dilakukan 5 kali pengujian sebagai bentuk evaluasi dari model yang telah dibuat. Dengan menggunakan *layer* yang sama yaitu 256, 256, 128, 128, 3.

Tabel 4. 1 Hasil Pengujian 1

epoch	Lr = 0,01			
	accuracy	precision	recall	f1-score
50	0,73	0,75	0,73	0,73
75	0,73	0,72	0,73	0,72
100	0,72	0,73	0,71	0,72
125	0,76	0,75	0,74	0,74
150	0,71	0,71	0,71	0,71

Pada tabel di atas adalah hasil dari proses pengujian pertama yang telah dilakukan pada sampel dataset yang sudah ada dengan *learning rate* 0,01. Maka didapatkan hasil yang paling tinggi ada pada *epoch* 125 dengan *accuracy* 76%, *precision* 75%, *recall* 74%, dan *f1-score* 74%.

Tabel 4. 2 Hasil Pengujian 2

epoch	Lr = 0,005			
	accuracy	precision	recall	f1-score
50	0,75	0,77	0,75	0,76
75	0,75	0,76	0,74	0,74
100	0,73	0,75	0,72	0,73
125	0,74	0,76	0,74	0,75
150	0,75	0,75	0,74	0,74

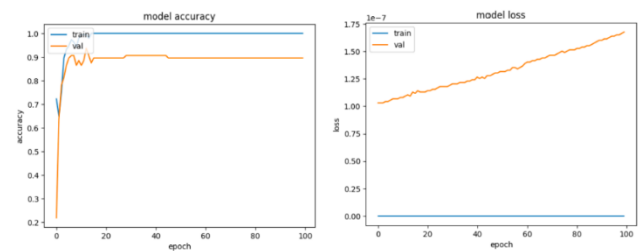
Pada tabel di atas adalah hasil dari proses pengujian kedua yang telah dilakukan pada sampel dataset yang sudah ada dengan *learning rate* 0,005. Maka didapatkan hasil yang paling tinggi ada pada *epoch* 50 dengan *accuracy* 75%, *precision* 77%, *recall* 75%, dan *f1-score* 76%.

Tabel 4. 3 Hasil Pengujian 3

epoch	Lr = 0,0001			
	accuracy	precision	recall	f1-score
50	0,73	0,74	0,74	0,71
75	0,77	0,78	0,77	0,77
100	0,75	0,75	0,75	0,75
125	0,76	0,76	0,76	0,76
150	0,74	0,75	0,75	0,74

Pada tabel di atas adalah hasil dari proses pengujian ketiga yang telah dilakukan pada sampel dataset yang sudah ada dengan *learning rate* 0,0001. Maka didapatkan hasil yang paling tinggi ada pada *epoch* 75 dengan *accuracy* 77%, *precision* 78%, *recall* 77%, dan *f1-score* 77%. Jadi dari 5 pengujian yang dilakukan, akurasi terbaik adalah 79% dengan tingkat pembelajaran 0,001.

Gambar 4.1 menampilkan grafik perbandingan akurasi dan kerugian. Grafik tersebut menggunakan warna biru untuk merepresentasikan validasi dan biru untuk merepresentasikan pelatihan. Jumlah *class* dan *epoch* yang berbeda dalam setiap *running* dapat mempengaruhi hasil ini.



Gambar 4.1 Grafik *accuracy* dan *loss*

V. KESIMPULAN

Dari hasil pengujian yang dilakukan sesuai prosedur yang telah ditentukan, diperoleh hasil yang berbeda seperti sangat matang, matang dan belum matang dari percobaan yang berbeda. Hal ini dapat terjadi karena ada beberapa faktor yang mempengaruhi seperti hormon auksin, giberelin, dan gas etilen pada masing-masing semangka

Keseluruhan hasil proyek mendapatkan hasil yang cukup baik, dari proses front end, back end, serta proses prediksi. Waktu yang dibutuhkan untuk keseluruhan proses sudah mendapatkan hasil di angka yang cepat yaitu 10 detik, lalu pada tujuan friendly usefull, sudah sangat dapat digunakan dengan mudah oleh semua kalangan masyarakat yang ingin menggunakannya. Namun terdapat kekurangan terutama pada klasifikasi belum matang, kurangnya dataset, kurangnya keandalan alat pendeteksi, ataupun kurangnya keahlian penjual dalam memberikan informasi dapat menjadi penyebab benar dikatakan salah pada alat pendeteksi.

REFERENSI

- [1] V. S.Thomas, S. Darvesh,C.MacKnight, and K.Rockwood, "Estimating the prevalenceof dementia in elderly people: a comparison of the Canadian Study of Health and Aging and National Population Health Survey approaches," Int Psychogeriatr, vol. 13 Supp 1,no. SUPPL. 1, pp. 169–175, 2001, doi: 10.1017/S1041610202008116.
- [2] M. M. Baig and H. Gholamhosseini, "Smart health monitoring systems: an overview ofdesign and modeling," J Med Syst, vol. 37, no. 2, Apr. 2013, doi: 10.1007/S10916-012-9898-Z.
- [3] M. M. Alam, H. Malik, M. I. Khan, T. Pardy, A. Kuusik, and Y. le Moullec, "A survey on the roles of communication technologies in IoT-Based personalized healthcare applications," IEEE Access, vol. 6, pp. 36611– 36631, Jul. 2018, doi: 10.1109/ACCESS.2018.2853148.
- [4] S. Li, L. da Xu, and X. Wang, "A continuous biomedical signal acquisition system basedon compressed sensing in body sensor networks," IEEE Trans Industr Inform, vol. 9, no. 3, pp. 1764–1771, 2013, doi: 10.1109/TII.2013.2245334.
- [5] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," IEEE J Biomed Health Inform, vol. 17, no. 3, pp. 579–590, 2013, doi: 10.1109/JBHI.2012.2234129.

- [6] A. Arcelus, R. Goubran, M. H. Jones, and F. Knoefel, "Integration of smart home technologies in a health monitoring system for the elderly," Proceedings - 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINAW'07, vol. 1, pp. 820–825, 2007, doi: 10.1109/AINAW.2007.209.
- [7] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based 12 systems for health monitoring and prognosis," IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 40, no. 1, pp. 1–12, 2010, doi: 10.1109/TSMCC.2009.2032660.
- [8] M. E. Garbelini et al., "SweynTooth: Unleashing Mayhem over Bluetooth Low Energy", Accessed: May 31, 2022. [Online]. Available: <https://www.usenix.org/conference/atc20/presentation/garbelini>
- [9] A. S. Seferagić et al., "Survey on Wireless Technology Trade-Offs for the Industrial Internet of Things," Sensors 2020, Vol. 20, Page 488, vol. 20, no. 2, p. 488, Jan. 2020, doi: 10.3390/S20020488.
- [10] V. S. Thomas, S. Darvesh, C. MacKnight, and K. Rockwood, "Estimating the Prevalence of Dementia in Elderly People: A Comparison of the Canadian Study of Health and Aging and National Population Health Survey Approaches," Int Psychogeriatr, vol. 13, no. S1, pp. 169–175, 2001, doi: 10.1017/S1041610202008116.
- [11] S. Majumder, T. Mondal, and M. J. Deen, "Wearable Sensors for Remote Health Monitoring," Sensors (Basel), vol. 17, no. 1, Jan. 2017, doi: 10.3390/S17010130.