IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 5, May 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305348 |

Deep Voice: Real-Time Neural Text-to-Speech

Pranay Narule, Mr. Vijay Rakhade, Mr. Lowlesh Yadav

Student, Department of Computer Science & Engineering Shri Sai college of Engineering & Technology, Chandrapur, India

Assistant Professor, Department of Computer Science & Engineering Shri Sai college of Engineering & Technology, Chandrapur, India

Assistant Professor, Department of Computer Science & Engineering Shri Sai college of Engineering & Technology, Chandrapur, India

ABSTRACT: We present Deep Voice, a production-quality text-to-speech system constructed entirely from deepneural networks. Deep Voice lays the groundwork for truly end-to-end neural speech synthesis. The system comprises five major building blocks: a segmentation model for locating phoneme boundaries, a grapheme-tophoneme conversion model, a phoneme duration prediction model, a fundamental frequency prediction model, and an audio synthesis model. For the segmentation model, we propose a novel way of performing phoneme boundary detection with deep neural networks using connectionist temporal classification (CTC) loss. For the audio synthesis model, we implement a variant of WaveNet that requires fewer parameters and trains faster than the original. By using a neural network for each component, our system is simpler and more flexible than traditional text-tospeech systems, where each component requires laborious feature engineering and extensive domain expertise. Finally, we show that inference with our system can be performed faster than real time and describe optimized WaveNet inference kernels on both CPU and GPU that achieve up to 400x speedups over existing implementations.

KEYWORDS: Deep Voice, text-to-speech system, WaveNe, Intelligent ProcessAutomation.

I. INTRODUCTION

Synthesizing artificial human speech from text, commonly known as text-to-speech (TTS), is an essential component in many applications such as speech-enabled devices, navigation systems, and accessibility for the visually-impaired. Fundamentally, it allows human-technology interaction without requiring visual interfaces. Modern TTS systems are based on complex, multi-stage processing pipelines, each of which may rely on hand-engineered features and heuristics. Due to this complexity, developing new TTS systems can be very labor intensive and difficult. Deep Voice is inspired by traditional text-to-speech pipelines and adopts the same structure, while replacing all components with neural networks and using simpler features: first we convert text tophoneme and then use an audio synthesis model to convert linguistic features into speech (Taylor, 2009). Unlike prior work (which uses hand-engineered features such as spectral envelope, spectral parameters, aperiodic parameters, etc.), our only features are phonemes with stress annotations, phoneme durations, and fundamental frequency (F0). This choice of features makes our system more readily applicable to new datasets, voices, and domains without any manual data annotation or additional feature engineering. We demonstrate this claim by retraining our entire pipeline without any hyperparameter changes on an entirely new dataset that contains solely audio and unaligned textual transcriptions and generating relatively high quality speech. In a conventional TTS system this adaptation requires days to weeks of tuning, whereas Deep Voice allows you to do it in only a few hours of manual effort and the time it takes models to train.

Real-time inference is a requirement for a productionquality TTS system; without it, the system is unusable for most applications of TTS. Prior work has demonstrated that a WaveNet (van den Oord et al., 2016) can generate close to human-level speech. However, WaveNet inference posesa daunting computational problem due to the high-frequency, autoregressive nature of the model,and it has been hitherto unknown whether such models can be used in a production system. We answer this question in the affirmative and demonstrate efficient, faster-than-real-time WaveNet inference kernels that produce high-quality 16 kHz audio and realize a 400X speedup over previous WaveNet inference implementations (Paine et al., 2016).

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305348 |

vendors of RPA tools report a surge in demand since 2016 [3], and we see some research where these tools are used for automating digital forensics [4], auditing [5] and industry [6]. The advent of the fourth industrial revolution (Industry 4.0) is paving the way for new ways to automate mundane rules-based business processes, using RPA tools on information obtained from smart devices [6]. For business processes, RPA is the extrapolation of a human worker's repetitive tasks by a robot (where those tasks are done quickly and profitably). This aims to replace people by automation in an outside-in manner. Unlike traditional methods, RPA is not part of the information infrastructure but rather sits on top of it, implying a low level of intrusiveness [7] possibly reducing costs. Some reports present a 30% to 50% decrease in operational costs of transactional activities within shared services with the use of RPA technologies [8]

II. METHODOLOGY

Previous work uses neural networks as substitutes for several TTS system components, including grapheme-to phoneme conversion models (Rao et al., 2015; Yao & Zweig, 2015), phoneme duration prediction models (Zen & Sak, 2015), fundamental frequency prediction models (Pascual & Bonafonte, 2016; Ronanki et al., 2016), and audio synthesis models (van den Oord et al., 2016;Mehri et al., 2016). Unlike Deep Voice, however, none of these systems solve the entire problem of TTS and many of them use specialized hand-engineered features developed specifically for their domain. Most recently, there has been a lot of work in parametric audio synthesis, notably WaveNet, SampleRNN, and Char2Wav (van den Oord et al., 2016; Mehri et al., 2016; Sotelo et al., 2017). While WaveNet can be used for both conditional and unconditional audio generation, SampleRNN is only used for unconditional audio generation. Char2Wav extends SampleRNN with an attention-based phoneme duration model and the equivalent of an F0 prediction model, effectively providing local conditioning information to a SampleRNN-based vocoder. Deep Voice differs from these systems in several key aspects that notably increase the scope of the problem. First, Deep Voice is completely standalone; training a new Deep Voice system does not require a pre-existing TTS system, and can be done from scratch using a dataset of short audio clips and corresponding textual transcripts. In contrast, reproducing either of the aforementioned systems requires access and understanding of a pre-existing TTS system, because they use features from another TTS system either at training or inference time.

Second, Deep Voice minimizes the use of hand-engineered features; it uses one-hot encoded characters for grapheme to phoneme conversion, one-hot encoded phonemes and stresses, phoneme durations in milliseconds, and normalized log fundamental frequency that can be computed from waveforms using any F0 estimation algorithm. All of these can easily be obtained from audio and transcripts with minimal effort. In contrast, prior works use a much more complex feature representation, that effectively makes reproducing the system impossible without a pre- existing TTS system. WaveNet uses several features from a TTS system (Zen et al., 2013), that include values such as the number of syllables in a word, position of syllables in the phrase, position of the current frame in the phoneme, and dynamic features of the speech spectrum like spectral and excitation parameters, as well as their time derivatives. Char2Wav relies on vocoder features from the WORLD TTS system (Morise et al., 2016) for pre-training their alignment module which include F0, spectral envelope, and aperiodic parameters. Finally, we focus on creating a production-ready system, which requires that our models run in real-time for inference.Deep Voice can synthesize audio in fractions of a second, and offers a tunable trade-off between synthesis speed and audio quality. In contrast, previous results with WaveNet require several minutes of runtime to synthesize one second of audio. We are unaware of similar benchmarks for SampleRNN, but the 3-tier architecture as described in the original publication requires approximately 4-5X as much compute during inference as our largest WaveNet models, so running the model in real-time may prove challenging

III. TTS SYSTEM COMPONENTS

As shown in Fig. 1,
the TTS system consists of five major building blocks:

- The grapheme-to-phoneme model converts from written text (English characters) to phonemes (encoded using a phonemic alphabet such as ARPABET).
- The segmentation model locates phoneme boundaries in the voice dataset. Given an audio file and a phoneme-by-phoneme transcription of the audio, the segmentation model identifies where in the audio each phoneme begins and ends.
- The phoneme duration model predicts the temporal duration of every phoneme in a phoneme sequence (an utterance).
- The fundamental frequency model predicts whether a phoneme is voiced. If it is, the model predicts the fundamental frequency (F0) throughout the phoneme's duration.
- The audio synthesis model combines the outputs of the grapheme-to-phoneme, phoneme duration, and

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305348 |

fundamental frequency prediction models and synthesizes audio at a high sampling rate, corresponding to

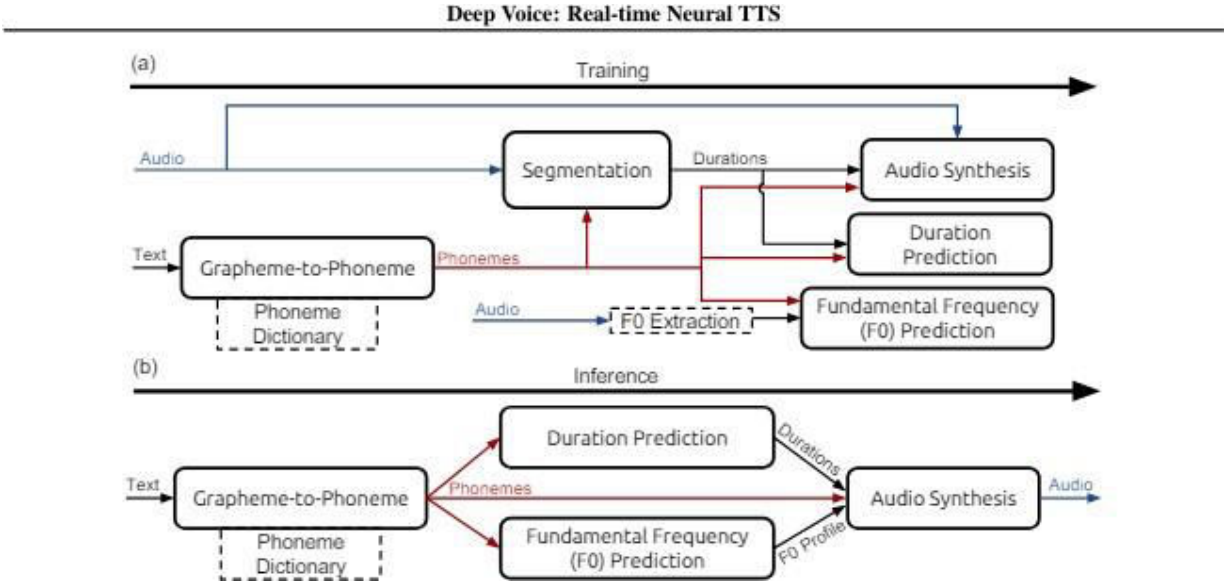

Figure 1. System diagram depicting (a) training procedure and (b) inference procedure, with inputson the left and outputs on the right. In our system, the duration prediction model and the F0 prediction model are performed by a single neural network trained with a joint loss. The grapheme-to-phoneme model is used as a fallback for words that are not present in a phoneme dictionary, suchas CMUDict. Dotted lines denote non-learned components.

The desired text. During inference, text is fed through the grapheme-tophoneme model or a phoneme dictionary to generate phonemes. Next, the phonemes are provided as inputs to the phoneme duration model and F0 prediction model to assign durations to each phoneme and generate an F0 contour. Finally, the phonemes, phoneme durations, and F0 are used as local conditioning input features to the audio synthesis model, which generates the final utterance. Unlike the other models, the segmentation model is not used during inference. Instead, it is used to annotate the training voice data with phoneme boundaries. The phoneme boundaries imply durations, which can be used to train the phoneme duration model. The audio, annotated with phonemes and phoneme durations as well as fundamental frequency, is used to train the audio synthesis model. In the following sections, we describe all the building blocks in detail.

3.1 Grapheme-to-Phoneme Model

Our grapheme-to-phoneme model is based on the encoderdecoder architecture developed by (Yao & Zweig, 2015). However, we use a multi-layer bidirectional encoder with a gated recurrent unit (GRU) nonlinearity and an equally deep unidirectional GRU decoder (Chung et al., 2014). The initial state of every decoder layer is initialized to the final hidden state of the corresponding encoder forward layer. The architecture is trained with teacher forcing and decoding is performedusing beam search. We use 3 bidirectional layers with 1024 units each in the encoder and 3 unidirectional layers of the same size in the decoder and a beam search with a width of 5 candidates. During training, we use dropout with probability 0.95 after each recurrent layer. For training, we use the Adam optimization algorithm with $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varepsilon = 10{-}8$, a batch size of 64, a learning rate of $10{-}3$, and an annealing rate of 0.85 applied every 1000 iterations (Kingma & Ba, 2014).

All these steps help the team design the To-Be state of truly digital automated processes with a business case clearly articulating the ROI with timelines. The project manager documents and floats a detailed project plan with mutually agreed (between client-vendor) timelines in order to assign responsibilities as per the RACI matrix. Documentation and creation of artefacts, which canbe used and re-used for later purposes, is necessary at this stage.

The objectives at this stage are to create the keystroke level To-Be maps, business case, and detailed project plan with clearly called-out responsibilities.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305348 |

3.2 Segmentation Model

Our segmentation model is trained to output the alignment between a given utterance and a sequence of target phonemes. This task is similar to the problem of aligning speech to written output in speech recognition. In that domain, the connectionist temporal classification (CTC) lossfunction has been shown to focus on character alignments to learn a mapping between sound andtext (Graves et al., 2006). We adapt the convolutional recurrent neural network architecture from a state-of-the-art speech recognition system (Amodei et al., 2015) for phoneme boundary detection. A network trained with CTC to generate sequences of phonemes will produce brief peaks for every output phoneme. Although this is sufficient to roughly align the phonemes to the audio, it is insufficient to detect precise.

Real-time Neural TTS phoneme boundaries. To overcome this, we train to predict sequences of phoneme pairs rather than single phonemes. The network will then tend to output phoneme pairs at timesteps close to the boundary between two phonemes in a pair. To illustrate our label encoding, consider the string “Hello!”. To convert this to a sequence of phoneme pair labels, convert the utterance to phonemes (using a pronunciation dictionary such as CMUDict or a grapheme-tophoneme model) and pad the phoneme sequence on either end with the silence phoneme to get “sil HH EH L OW sil”. Finally, construct consecutive phoneme pairs and get “(sil,HH), (HH, EH), (EH, L), (L, OW), (OW, sil)”. Input audio is featurized by computing 20 Mel- frequency cepstral coefficients (MFCCs) with a ten millisecond stride. On top of the input layer, there are two convolution layers (2D convolutions in time and frequency), three bidirectional recurrent GRU layers, and finally a softmax output layer. The convolution layers use kernels withunit stride, height nine (in frequency bins), and width five (in time) and the recurrent layers use 512 GRU cells (for each direction). Dropout with a probability of 0.95 is applied after the last convolution and recurrent layers. To compute the phoneme-pair error rate (PPER), we decode using beam search. To decode phoneme boundaries, we perform a beam search with width 50 withthe constraint that neighboring phoneme pairs overlap by at least one phoneme and keep track of the positions in the utterance of each phoneme pair. For training, we use the Adam optimization algorithm with $\beta1 = 0.9$, $\beta2 = 0.999$, $\varepsilon = 10^{-8}$, a batch size of 128, a learning rate of 10^{-4} , and an annealing rate of 0.95 applied every 500 iterations (Kingma & Ba, 2014).

3.3 Phoneme Duration and Fundamental Frequency

Model We use a single architecture to jointly predict phoneme duration and time-dependent fundamental frequency. The input to the model is a sequence of phonemes with stresses, with eachphoneme and stress being encoded as a one-hot vector. The architecture comprises two fully connected layers with 256 units each followed by two unidirectional recurrent layers with 128 GRU cells each and finally a fullyconnected output layer. Dropout with a probability of 0.8 is applied after the initial fully-connected layers and the last recurrent layer. The final layer producesthree estimations for every input phoneme: the phoneme duration, the probability that the phoneme is voiced (i.e. has a fundamental frequency), and 20 time-dependent F0 values, which are sampleduniformly over the predicted duration. The model is optimized by minimizing a joint loss that combines phoneme duration error, fundamental frequency error, the negative log likelihood of theprobability that the phoneme is voiced, and a penalty term proportional to the absolute change of F0 with respect to time to impose smoothness. The specific functional form of the loss function isdescribed in Appendix B. For training, we use the Adam optimization algorithm with $\beta1 = 0.9$, $\beta2 = 0.999$, $\varepsilon = 10^{-8}$, a batch size of 128, a learning rate of 3×10^{-4} , and an annealing rate of 0.9886applied every 400 iterations (Kingma & Ba, 2014). 3.4. Audio Synthesis Model Our audio synthesis model is a variant of WaveNet. WaveNet consists of a conditioning network, which upsamples linguistic features to the desired frequency, and an autoregressive network, which generates a probability distribution P(y) over discretized audio samples $y \in \{0, 1, \ldots, 255\}$. We vary the number of layers `, the number of residual channels r (dimension of the hidden state of every layer), and the number of skip channels s (the dimension to which layer outputs are projectedprior to the output layer). WaveNet consists of an upsampling and conditioning network, followedby ` 2×1 convolution layers with r residual output channels and gated tanh nonlinearities. We breakthe convolution into two matrix multiplies per timestep with Wprev and Wcur. These layers are connected with residual connections. The hidden state of every layer is concatenated to an `r vectorand projected to s skip channels with Wskip, followed by two layers of 1×1 convolutions (with weights Wrelu and Wout) with relu nonlinearities. WaveNet uses transposed convolutions for upsampling and conditioning. We find that our models perform better, train faster, and require fewer parameters if we instead first encode the inputs with a stack of bidirectional quasi-RNN (QRNN) layers (Bradbury et al., 2016) and then perform upsampling by repetition to the desired frequency. Our highest-quality final model uses ` = 40 layers, r = 64 residual channels, and s = 256skip channels. For training, we use the Adam optimization algorithm with $\beta1 = 0.9$, $\beta2 = 0.999$, $\varepsilon = 10^{-8}$, a batch size of 8, a learning rate of 10^{-3} , and an annealing rate of 0.9886 applied every 1,000 iterations (Kingma & Ba, 2014). Please refer to Appendix A for full details of our WaveNetarchitecture and the QRNN layers we use.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305348 |

IV. CONCLUSION

In this work, we demonstrate that current Deep Learning approaches are viable for all the components of a highquality text-to-speech engine by building a fully neural system. We optimizeinference to faster-than-real-time speeds, showing that these techniques can be applied to generateaudio in real-time in a streaming fashion. Our system is trainable without any human involvement, dramatically simplifying the process of creating TTS systems. Our work opens many new possibledirections for exploration. Inference performance can be further improved through careful optimization, model quantization on GPU, and int8 quantization on CPU, as well as experimentingwith other architectures such as the Xeon Phi. Another natural direction is removing the separation between stages and merging the segmentation, duration prediction, and fundamental frequency prediction models directly into the audio synthesis model, thereby turning the problem into a full sequence-to-sequence model, creating a single endto-end trainable TTS system, and allowing us to train the entire system with no intermediate supervision. In lieu of fusing the models, improving the duration and frequency models via larger training datasets or generative modeling techniques may have an impact on voice naturalness.

REFERENCES

1. Infopédia (2020). Dicionário Infopédia da Língua Portuguesa, 2020. [Online]. Available from : https://www.infopedia.pt.
2. van der Aalst, W. M., Bichler, M., & Heinzl, A. (2018). Robotic Process Automation. Bus InfSyst Eng 60, pp.269–272. https://doi.org/10.1007/s12599-018-0542-4
3. Enríquez, J. G., Jiménez-Ramírez, A., Domínguez-Mayo, F. J., & García-García, J. A. (2020). Robotic Process Automation: A Scientific and Industrial Systematic Mapping Study. IEEEAccess, 8, 39113-39129.
4. Williams, D., & Allen, I. (2017). Using artificial intelligence to optimize the value of robotic process automation. Available from: https://www.ibm.com/downloads/cas/KDKAAK29 [9] Nilsson, N. J. (2014). Principles of artificial intelligence. Morgan Kaufmann Editors .
5. Coccia, M. Asymmetry of the technological cycle of disruptive innovations. Technol. Anal.Strat. Manag. **2020**, 32, 1462–1477.
6. [CrossRef]
7. Bygstad, B., & Iden, J. (2017). A governance model for managing lightweightIT. In Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, & S.
8. Costanzo (Eds.), Recent advances in information systems and technologies (pp. 384–393). Cham: Springer International Publishing.https://doi.org/10.1007/978-3-319-56535-4_39.
9. Zheng, P., Sang, Z., Zhong, R. Y., Liu, Y., Liu, C., Mubarok, K., ... & Xu, X. (2018). Smartmanufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives. Frontiers of Mechanical Engineering, 13(2), pp:137-150.
10. Ustundag, A., & Cevikcan, E. (2017). Industry 4.0: managing the digital transformation.Springer Editors. Available from: https://www.springer.com/gp/book/9783319578699
11. Haenlein, Michael & Kaplan, Andreas. (2019). A Brief History of Artificial Intelligence: Onthe Past, Present, and Future of Artificial Intelligence. California Management Review.
12. Mitchell, T. M. (1997). Machine Learning. New York: McGraw-Hill. ISBN: 978-0-07-042807-2.
13. Leno, V., Dumas, M., La Rosa, M., Maggi, F. M., & Polyvyanyy, A. (2020). Automated Discovery of Data Transformations for Robotic Process Automation. https://arxiv.org/abs/2001.01007
14. Huang, F., & Vasarhelyi, M. A. (2019). Applying robotic process automation (RPA) in auditing: A framework. INTERNATIONAL JOURNAL OF ACCOUNTING INFORMATION SYSTEMS, 35. https://doi.org/10.1016/j.accinf.2019.100433
15. Agostinelli, S., Marrella, A., & Mecella, M. (2020). Towards Intelligent Robotic ProcessAutomation for BPMers. Available from: https://www.researchgate.net/publication/338401505_Towards_Intelligent_Robotic_Process_Automation_for_BPMers
16. FLUSS, D. (2018). Smarter Bots Mean Greater Innovation, Productivity, and Value: Robotic process automation is allowing companies to re-imagine and re-invest in all aspects oftheir businesses. CRM Magazine, 22(10), 38–39.
17. Delloite (2019). Automation with intelligence Reimagining the organisation in the 'Age of With'. Available from: https://www2.deloitte.com/content/dam/Deloitte/tw/Documents/strategy/tw-Automation-with- intelligence.pdf
18. Lowlesh Yadav and Asha Ambhaikar, "IOHT based Tele-Healthcare Support System for Feasibility and performance analysis," Journal of Electrical Systems, vol. 20, no. 3s, pp. 844–850, Apr. 2024, doi: 10.52783/jes.1382.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305348 |

19. L. Yadav and A. Ambhaikar, "Feasibility and Deployment Challenges of Data Analysis in Tele-Healthcare System," 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), Raipur, India, 2023, pp. 1-5, doi: 10.1109/ICAIIHI57871.2023.10489389.
20. L. Yadav and A. Ambhaikar, "Approach Towards Development of Portable Multi-ModelTele-Healthcare System," 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), Raipur, India, 2023, pp. 1-6, doi: 10.1109/ICAIIHI57871.2023.10489468.
21. Lowlesh Yadav and Asha Ambhaikar, Exploring Portable Multi-Modal Telehealth Solutions: A Development Approach. International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), vol. 11, no. 10, pp. 873–879, Mar. 2024.11(10), 873–879, DOI: 10.13140/RG.2.2.15400.99846.
22. Lowlesh Yadav, Predictive Acknowledgement using TRE System to reduce cost and Bandwidth, March 2019. International Journal of Research in Electronics and Computer Engineering (IJRECE), VOL. 7 ISSUE 1 (JANUARY- MARCH 2019) ISSN: 2393-9028(PRINT) | ISSN: 2348-2281 (ONLINE).

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details