Khandokar Md. Nayem

nayem.github.io

+1 (812) 369-6358
knayem@iu.edu

Computer science Ph.D. student doing research on machine learning and deep network architectures on speech applications. Experienced in working on speech recognition, natural language processing (NLP), computer vision for around 4 years of experience in the software industry, and academia.

EDUCATION

Ph.D. in Computer Science, May 2022 (Anticipated)
Indiana University, Bloomington, IN — Advisor: Prof. Donald S. Willamson

M.Sc. in Computer Science, December 2019
Indiana University, Bloomington, IN

B.Sc. in Computer Science & Engineering (CSE), July 2014
Bangladesh University of Engineering & Technology (BUET), Dhaka, Bangladesh

EMPLOYMENT

Indiana University, Bloomington, IN, *Research Assistant, ASPIRE research lab* — Fall 2016 - Present

- Researched noise cancellation techniques to filter out a wide range of noises from human speech. Also developed a quantized end-to-end speech enhancement system and evaluated by human with survey. (Python, LSTM, Qualtrics)
- Formulated prediction models for nulliparous pregnant women to detect gestational diseases like preeclampsia and hypertension, and to trace the impact of physical activity collected by wearable devices. (Python, ANOVA, DNN)
- Investigated human emotion (arousal & valence) detection techniques in speech using deep neural models. (CNN, RNN)

BOSE Corporation, Boston, MA, *Machine Learning/Neural Signal Processing Intern* — Summer 2020

- Researched enhancing speech in remote microphones by self-speech removal to provide better quality sound with low latency to the hearing aids and voice-assistive wearable devices. (Python, TensorFlow, LSTM)

REVE Systems, Dhaka, *Jr. Software Engineer, Team Media Gateway* — January 2015

- Programmed media gateway controller to facilitate both calls and faxes between the telephone network and VoIP network or another telephone network. Also designed front-end panel for VoIP administrators and customers for easy use. (Back-end: protocol megaco 1.0, Java; front-end: .JSP framework, Javascript, Ajax, MySQL)

PUBLICATIONS

- **Khandokar Md. Nayem**, Donald S. Williamson, "Incorporating Embedding Vectors from a Human Mean-Opinion Score Prediction Model for Monaural Speech Enhancement", in Proc. INTERSPEECH 2021. (in review - paper)
- **Khandokar Md. Nayem**, Donald S. Williamson, "Towards an ASR approach using Acoustic and Language Models for Speech Enhancement", in Proc. ICASSP 2021. (paper, poster, slides, video)
- **Khandokar Md. Nayem**, Donald S. Williamson, "Monaural speech enhancement using intra-spectral recurrent layers in the magnitude and phase responses", in Proc. ICASSP 2020. (paper, slides, video)
- **Khandokar Md. Nayem**, Donald S. Williamson, "Incorporating Intra-Spectral Dependencies with a Recurrent Output Layer for Improved Speech Enhancement", in Proc. MLSP 2019. (paper, poster)
- Shujon Naha, **Khandokar Md. Nayem**, Md. Lisul Islam, "RSGAN: Recurrent Stacked Generative Adversarial Network for Conditional Video Generation", presented at IU computer vision project showcase, 2017. (paper, poster)

PROGRAMMING SKILLS

Languages	Python, C++, Matlab, C, Java, HTML, CSS, Javascript, R, SQL, Assembly, Shell Script
Frameworks & Libraries	TensorFlow, Keras, NLTK, CImg, Cmake, OpenCV, Scikit-Learn, OpenGL, Qualtrics
Machine Learning	Regression, Clustering, Decision Tree, Ensemble Methods, DNN, CNN, RNN, LSTM

SELECTED PROJECTS

- Speech enhancement and separation using auxiliary information like phonemic structure or textual information without alignment to boast performance and speech quality. (Python, TensorFlow, Attention model; code)
- Automatic classification of rhetorical questions with stress detection on our own collected dataset using Recurrent Neural Network (RNN) and Convolutional Recurrent Neural Network (CRNN) models. (Python, Keras, NLTK; code)
- An end-to-end speech recognition system for the English language using bi-directional recurrent neural network architecture without any frame wise labeling. (Python, Keras, NLTK; code)
- Image Matching to match images that are taken from different viewpoints of the same object; detect object like car from aerial snapshot; create a panoramic image stitching multiple images. (C++, CImg, OverFeat packages; code)