AYUSH SHRIDHAR

ayush.shridhar1999@gmail.com
(+91)9969462724

WORK EXPERIENCE

The Australian National University December'19 - June'20
Visiting Researcher, Research School of Computer Science *Canberra, ACT, Australia*
Advisor: Prof. Hanna Kurniawati

· Working on implementing partially observable markov decision process (POMDP) solvers to verify the assurance of driver-less cars. Language: C, C++, Python

University College London May'19 - August'19
Visiting Researcher, Department of Statistical Science and Machine learning *London, U.K*
Advisor: Prof. Franz Kiraly

· Working on a neural network port for MLJ: a machine learning toolbox being developed in UCL in collaboration with researchers from the Alan Turing Institute. The framework implements an interface to Flux.jl and supports advanced features such as learned entity embeddings. Language: Julia, C++

Mozilla May'19 - August'19
Google Summer of Code student *Remote*

· Contributing to bugbug : A platform for bugzilla machine learning projects. Implemented various deep learning and statistical models to make bug tracking on bugzilla simpler by automated issue labelling. Language: Python, Dockerfile

Bhabha Atomic Research Center November'18 - January'19
Research Intern, Super Computing Facility *Mumbai, India*

· Worked under the Data Science department on making face image classification system safe from adversarial attacks. Implemented generative models to detect fake images from the real ones. Language: Python, C++, CUDA-C

The Julia Language, NumFOCUS April'18 - August'18
Google Summer of Code student *Remote*

· Contributed to the FluxML ecosystem in Julia. Implemented state of the art models for computer vision tasks. Also wrote the ONNX.jl package completely from scratch. ONNX.jl provides functionalities to load high quality pre-trained deep learning models into Flux. Language: Julia

PROJECTS

ceesh

· *ceesh* is a shell written completely in C without using any advanced framework or library. It uses only standard library functions and supports advanced functionalities such as pipelines and running daemon processes. Languages: C

Keras.jl

· *Keras.jl* is a framework that loads pre-trained Keras models into Flux.jl. This is done by tracing the computation graph using an code intermediate representation tool *Dataflow.jl* and mapping each layer simultaneously into Flux operators.
Languages: Julia

kavernets

· *kaverents* provides an easy to use API to implement various Generative Adversarial Models. It is based on the Keras API and uses TensorFlow backend.
Languages: Python

EDUCATION

International Institute of Information Technology, Bhubaneswar, India *August 2016 - Present*
Bachelor's in Technology
Department of Computer Science

PUBLICATIONS

1. *In review*: "Interoperating Deep Learning models with ONNX.jl" *Proceedings of JuliaCon'19*
***Ayush Shridhar**, Michael Innes, Phil Tomson*

TECHNICAL SKILLS

Programming Languages: C/C++, Python, CUDA, JavaScript, Julia, HTML, Java, Cython, Shell Scripting
Operations Systems: Linux, Windows
Frameworks: TensorFlow, Keras, PyTorch, Flux.jl, Pandas, NumPy Matplotlib, OpenCV, Git, Django, D3.js, PostgreSQL

REFERENCES

- **Prof. Franz Kiraly**, Department of Statistical Science, University College London, United Kingdom. *f.kiraly@ucl.ac.uk*
- **Prof. Anthony Blaom**, Lecturer, University of Auckland, New Zealand. *Available on demand*
- **Prof. Ajaya Dash**, Department of Computer Science, International Institute of Information Technology, Bhubaneswar, India. *ajaya@iiit-bh.ac.in*
- **Mr. Phil Tomson**, Intel Research, Portland, Oregon. *Available on demand*
- **Mr. Marco Castelluccio**, Mozilla, London. *Available on demand*