Harish Rajagopal

Fourth Year Undergraduate

Computer Science and Engineering · Indian Institute of Technology Kanpur

 rharish101 · rharish@iitk.ac.in · +91-7318019201

Educational Qualifications

Degree	Year	Institution/Board	CGPA/%
B. Tech	2016 - Present	IIT Kanpur	9.7/10.0
Sr. Secondary	2016	Maharashtra H.S.C	90.46%
Secondary	2014	Maharashtra S.S.C	93.6%

Academic Achievements

- Secured 7 **A⋆** grades across 6 semesters.
- Secured 7 **A⋆** grades for exceptional performance across 6 semesters.
- Awarded **Academic Achievement Awards** for notable performance in 1st and 2nd years.
- Secured All India Rank of **185** in **JEE Advanced 2016**.
- Secured All India Rank of **185** in **JEE (Advanced) 2016** given by about 200 thousand students.
- Secured All India Rank of **205** in **JEE Mains 2016**.
- Secured All India Rank of **205** in **JEE (Mains) 2016** given by about 1.4 million students.

Internships

- **Research Intern, NYU Tandon** $\hfill$ May '19 - July '19
  Prof. Paweł Korus, Prof. Nasir Memon

  - Researched robust image hashes that are immune to typical image transformations, while being sensitive to malicious image edits such as face swaps, deep fakes, object addition/deletion.
  - Constructed models that take an image and output a binary hash that would give lower *Hamming distances* for hashes of similar images while giving more considerable distances for dissimilar images.
  - Developed a *framework* for testing against compression, contrast changes, gamma, blurring, warping.
  - Developed a *framework* for testing against transformations like compression, contrast changes, blurring, warping.
  - Developed a test framework to obtain metrics based on Hamming distances of hashes of images against common transformations such as JPEG compression, gamma correction, contrast adjustments, blurring, warping.
  - Trained various network architectures on the *triplet loss* along with *mining* of examples for improved training.
  - Trained various novel network architectures on the *triplet loss* along with *mining* of examples for improved training.
  - Trained Various novel network architectures on different loss functions, notably the *triplet loss*, along with *mining* of examples for improved training by targeting semi-hard examples, similar to *FaceNet*.
  - The baseline architecture was a C++ based non-learning model that uses the Discrete Cosine Transform (DCT) coefficients to calculate the hash, interfaced with Python using Cython.
  - Tested other architectures including one with MobileNet as the feature extractor and a fully-connected layer, another neural network to approximate the baseline model, and a third one to combine both of these.
  - Tested the networks against *adversarial attacks* such as *FGSM*, *Projected Gradient Descent*, *Boundary Attack*.
  - Carried out black-box attack against the baseline model using an approximate substitute architecture.
  - Advanced adversarial attacks and defences to make the models robust will be implemented.

- **Research Intern (Remote), NYU Tandon** $\hfill$ May '18 - July '18
  Prof. Yao Wang $\hfill$ https://github.com/rharish101/Plasticity-Networks

  - Researched *differentiable plasticity* for *domain transfer* in images using Convolutional Neural Networks.
  - Used differentiable plasticity for transfer learning in domain change by training model for classification on one dataset and then adjusting for classification on another dataset.
  - Tested three architectures on the *Street View House Numbers (SVHN)* dataset: standard ResNet, ResNet with plasticity on the fully-connected layer and ResNet with plasticity on all layers, using 20 and 56 layer variants.

- – Improved *efficiency* in the temporal update rule for the *Hebbian weights* by using *transpose convolution*.
- – Modified the standard temporal update rule for the Hebbian matrix (plastic weights) for convolutional layers, by using transpose convolution for an efficient method of obtaining the plasticity update.
- – Tested *bilinear interpolation* followed by convolution as an alternative to transpose convolution.
- – Accounted for downsampling of the image (due to the usage of stridden convolution) in the new update rule by using bilinear interpolation of the initial convolution output as the kernel.
- – Achieved improvement in classification accuracy, when adapting models from the SVHN dataset to MNIST.
- – Achieved notable improvement in classification accuracy using full plasticity, when adapting models trained on the SVHN dataset for the MNIST dataset.
- – Achieved notable improvement in classification accuracy for the 20 layer models using full plasticity, when adapting models trained on the SVHN dataset for the MNIST dataset.
- – Initial results indicate notable improvement in classification accuracy by usage of differentiable plasticity in convolutional layers for the 20 layer models, while negligible improvement was observed for the 56 layer models.

- **Intern, Machine Learning Team, New York Office of IIT Kanpur** May '17 - July '18
  Prof. Manindra Agrawal

  - – Used the RAKE algorithm to assign word importance scores to each word in the short RSS summary and use them for a weighted average of Google News *Word2Vec* word vectors for the news' vector.
  - – Developed an *online* text clustering model using a fully-online modification of the *DBSCAN* algorithm.
  - – Created an online clustering model for the news' vector to identify articles having follow-up articles, using a modification of *DBSCAN* optimized for discrete & bounded data, to updates clusters on a per-article basis.
  - – Used a *tf-idf* weighted *time series analysis* of article words to assign a trend score to top articles.
  - – Integrated *Apache Kafka* with the model for efficient stream implementations of the input-output pipeline.
  - – Created a *Docker* image of the combined model for online deployment.

  - – Implemented an *online* document vectorisation model using *Distributed Memory paragraph vectors*.
  - – Implemented an *online* document vectorisation model based on the *Distributed Memory paragraph vectors* model.
  - – Created an online *Doc2Vec* model based on the *Distributed Memory* model, initially trained offline, then deployed.
  - – Used the *RAKE* algorithm to assign word importance scores to each word in the document and use them for a weighted average of *GloVe* word vectors for the document vector, later concatenated with the *Doc2Vec* vector.
  - – Integrated *Apache Kafka* with the model for efficient stream implementations of the input-output pipeline.
  - – Created a *Docker* image of the combined model for online deployment.

  - – Studied scraping websites in Python, using the BeautifulSoup library to parse HTML.
  - – Studied threading in Python for sending parallel web requests, and created scrapers to scrape financial websites.
  - – Created scrapers for various websites to obtain content and various metadata.
  - – Studied the CouchDB and Couchbase NOSQL databases and stored scraped data on a Couchbase database.
  - – Created a program to use these scrapers to obtain content from different websites, while avoiding duplicates by using a Redis server, and store them internally for later processing.
  - – Deployed above models using Docker and integrated with existing infrastructure using Apache Kafka.

  - – Developed a *Word2Vec* model to identify duplicate documents using *Word Mover's Distance* on word vectors.
  - – Studied the *Word2Vec* model for converting words in a vocabulary into vectors.
  - – Created a *Word2Vec* model using the python library *Gensim* using the *continuous bag-of-words (CBOW)* model.
  - – Trained the *Word2Vec* model, using content from the articles scraped, to predict the next word using n-grams.
  - – Compared article contents using *Word Mover's Distance* with the trained *Word2Vec* model.
  - – Using HTTP requests in Python, created a program that incorporates the *Word2Vec* model and *Word Mover's Distance* to obtain articles from a website stream and append original articles on a stream of trending content.

  - – Trained a Convolutional Neural Network with sliding windows for English Optical Character Recognition.
  - – Trained a Convolutional Neural Network with sliding windows for English Optical Character Recognition (OCR).
  - – Studied Convolutional Neural Networks for learning from images.
  - – Used *OpenCV* to augment the dataset using elastic deformations, rotations in images, slimming of the characters.
  - – Implemented sliding windows using the image pyramids to scan images for characters and remove multiple windows on a character using non-maximum suppression.
  - – Moved to using the *Tesseract* OCR engine for the task, though preprocessing of images was required.
  - – Using adaptive Gaussian thresholding for images with less noise, and absolute thresholding for noisy images.

PROJECTS

- **Improving GANs through Test-Time Constraints** — Jan '19 - Present
  Prof. Vinay Namboodiri and Prof. Chetan Arora — https://cse.iitk.ac.in/users/rharish/ugp2
  - Pre-trained *Generative Adversarial Networks* (GANs) are fine-tuned using *interactive user input*, inspired by the iGAN paper and the paper Exploiting Test Time Evidence to Improve Predictions of Deep Neural Networks.
  - The user provides sketches of edges on a single output of the generator, and a *difference-of-Gaussians* based loss is backpropagated through the generator to fine-tune the generator's weights.
  - A regularisation term in the loss prevents the weights from deviating away from the weights learned initially.

- **Multi-Agent GANs for Image Super-Resolution** — Aug '18 - Dec '18
  Prof. Vinay Namboodiri — https://cse.iitk.ac.in/users/rharish/ugp1
  - A *Multi-agent generalisation* of SRGAN inspired by MADGANs for image super-resolution in *TensorFlow*.
  - Four generators get the four corner sections of the input, and their outputs are joined to get the final image.
  - Four generators (with shared lower layers) get the four corner sections of the input (with a slight overlap), and their outputs are joined (negating the overlap) to get the final high-resolution image.
  - Each generator pairs with a discriminator, while a global discriminator acts on the final output.

- **Higher-Order Optimisation in Deep Learning** — Sept '18 - Nov '18
  Prof. Piyush Rai, CS771A Course Project — https://github.com/rharish101/CS771-Project
  - A survey on the use of *quasi-Newton methods* in deep learning as part of a course.
  - Surveyed *Hessian-Free optimisation*, *AdaQN*, and *Sum of Functions Optimiser (SFO)*.
  - Benchmarked Hessian-Free optimisation on an MLP against the *Adam* and SGD optimisers in *TensorFlow*.

- **7th Inter-IIT Tech Meet (Silver Medal)** — Dec '18
  IIT Kanpur Contingent — https://github.com/rharish101/eye-in-the-sky
  - The Eye in the Sky: Semantic segmentation of satellite images using a dataset of only 14 images.
  - Won *2nd place* using the *P-Net* architecture, trained on an augmented dataset generated by slicing each image into multiple images and using rotations.
  - Tuned model hyperparameters using the *Tree of Parzen Estimators* method from the *Hyperopt* library.

- **Compiler for Golang in Python** — Jan '19 - Apr '19
  Prof. Amey Karkare, CS335A Course Project — https://github.com/rharish101/CS335A
  - Developed a compiler for translating Golang to MIPS written in Python as part of a course.
  - A compiler for translating Golang to MIPS written in Python as part of a course.
  - Implemented basic C-like features like data types, functions, pointers, structs, library imports, and I/O.
  - Implemented basic C-like features like data types, variables, expressions, control statements (if-else, switch, loops), arrays, functions, pointers, structs, library imports, and I/O.
  - Implemented advanced features like composite literals, struct embeddings, typedefs/aliases, operator overloading, multiple value returns, multiple parallel assignments, short declarations, and short-circuit evaluation.

- **GemOS - Operating Systems Development** — Aug '18 - Nov '18
  Prof. Debadatta Mishra, CS330A — https://github.com/rharish101/CS330-Assignments
  - Developed an object-store FUSE filesystem.
  - Developed an object-store FUSE filesystem for storing files inside a single directory.
  - Implemented process scheduling, sleeping, signal handling, syscalls and exception handlers.
  - Implemented process scheduling and swapping, context creation, sleeping, and signal handling for GemOS.
  - Implemented syscalls for writing & expanding/shrinking memory, and handlers for page faults & divide-by-zero.
  - Implemented virtual memory and paging.
  - Implemented virtual memory and paging using a 4-depth radix tree page table.

- **No-Frills Cab Locator - Android App** — Sept '18 - Nov '18
  Prof. Nisheeth Srivastava, CS252A Course Project — https://nfcl.pythonanywhere.com
  - A one-button-touch cab service with apps created using the *Ionic framework* for Android as part of a course.
  - Developed separate customer and driver apps using *Angular2* in Typescript and *Sass* for styling.
  - Setup a slim *Django* server to find nearest drivers, with an SQL database to store available drivers.

- **6th Inter-IIT Tech Meet** Dec '17 - Jan '18
  IIT Kanpur Contingent https://cse.iitk.ac.in/users/rharish/sixth-tech-meet

- **Reinforcement Learning in Atari Games** Jan '17 - July '17
  Association of Computing Activities, IIT Kanpur https://github.com/rharish101/ACA-Project

- **Depression Therapy Chatbot** May '17 - July '17
  Programming Club, IIT Kanpur https://github.com/rharish101/PClub-Project

- **Visual Attention in Image Captioning (Mentored)** May '18 - July '18
  Programming Club, IIT Kanpur https://github.com/DEVANSH99/Image_cptning2018

TECHNICAL SKILLS

- **Programming Languages:** Python, Bash, C, C++, LaTeX, PHP, HTML+CSS, MySQL, Typescript
- **Software and Utilities:** TensorFlow, PyTorch, Keras, Numpy, Git, OpenCV, Hyperopt, Gensim, Ionic, AutoCAD

RELEVANT COURSES

Visual Recognition	Computational Cognitive Science	Compiler Design (A⋆)
Introduction to Machine Learning	Operating Systems	Theory of Computation
Algorithms II	Computing Laboratory - II (A⋆)	Computing Laboratory - I
Data Structures and Algorithms	Probability and Statistics	Discrete Mathematics
Propositional and First Order Logic	Introduction to Abstract Algebra	Introduction to Electronics
Introduction to Linear Algebra (A⋆)	Multivariate Calculus	Fundamentals of Computing (A⋆)

POSITIONS OF RESPONSIBILITY

- **Secretary**, Programming Club, IIT Kanpur Aug '17 - Apr '18
- **Academic Mentor**, Counselling Service, IIT Kanpur Mar '17 - Feb '18