Kundan Kumar

Senior Undergraduate | Computer Sc. and Engg. | IIT Kanpur

G219, Hall-1, IIT Kanpur
+91-7897584034 | kundankumar2510@gmail.com

EDUCATIONAL QUALIFICATIONS

Year	Degree/Board	Institution	CGPA/%
$2012 - present$	B.Tech	IIT Kanpur	9.6/10
2012	Senior Secondary, CBSE	Eklavya Educational Complex, Patna	92.8%
2010	Secondary, CBSE	Eklavya Educational Complex, Patna	9.6/10

SCHOLASTIC ACHIEVEMENTS

- Won **1st** prize in Xerox Research Innovation Challenge — 2015
- Secured **All India Rank 98** in *IIT JEE* among 500,000 candidates — 2012
- Represented IIT Kanpur in 2nd inter-IIT tech meet in software development competition and won 2nd prize for the institute. Awarded Certificate of Appreciation by Director, IIT Kanpur on the Republic Day for the same. — 2013-14
- Certificate of Merit for **Academic Excellence** for two consecutive years — 2012-13 & 2013-14
- Selected as a **Course Tutor** for ESC101 course (Fundamentals of computing) course at IIT Kanpur — 2015

INTERNSHIPS

Patch2vec: Finding semantic vector embeddings of image patches — *May-July, 2015*

Computer Vision Lab, University of Massachusetts Amherst — *Advisor: Subhransu Maji*

- Worked on finding vector representation of image patches such that semantically related patches lie close to each other in the resulting representation space. (An image patch is defined as some small rectangular region in the image.)
- Proposed to use co-occurrence of patches in a context as a metric for similarity and relatedness
- Divided each image in four non-overlapping rectangular region with each rectangle as a context. Formulated the objective function as the likelihood of co-occurrence of patches.
- Implemented stochastic gradient descent for getting optimum representation of image patches. (Souce code)

Using deep learning to automatically diagnose diseases based on clinical history of patients — *December, 2015*

Xerox Research Innovation Challenge

- Developed models to diagnose a set of 16 diseases using static (age, gender, ethnicity) and time series (vitals, lab tests, medicines prescribed, etc.) data of patients.
- Our two models - LSTM with raw features and gradient boosted decision trees with time series features - outperformed entries by all other teams

KEY PROJECTS

Visual Question Answering: A neural approach — *Autumn 2015*

- Developed an automated answering tool for open-ended and free-form questions based on images using **neural networks**
- Devised a novel cost function to take advantage of semantic regularity in word-embedding space and used it to convert classification over words task to regression over word vector spaces
- This helped in dealing with very large vocabulary and has potential generalisation ability over unseen words
- Wrote **GPU** optimized code using **Torch** library. Source code may be found here. Project report may be found here

Relation extraction and knowledge graph construction using Large Text data — *Spring 2015*

- Proposed and implemented algorithm for expanding the set of examples for a concept/entity using semantic word vector embeddings
- Proposed another algorithm for relation (between different concepts) extraction using the semantic consistency of vector offsets in the embedding space
- Initial experiments on country-capital and country-language relationships gave precision of nearly 68.4%

Sentiment Prediction for movie reviews — *Spring 2015*

- Work on developing an automatic sentiment(positive or negative) classifier for movie reviews
- Used learning algorithms like SVM, random forests and logistic regression for learning models with various features e.g. word vectors, TFIDF, sentowordnet scores, z1-score, etc.
- Achieved best performance of 0.956 area under ROC curve with TFIDF features and SVM with linear kernel

Pascal to MIPS assembly compiler — *Spring 2015*

- Used **PLY (Python Lex-Yacc)** for writing lexer and semantic rules (parser) to generate three-address code
- Implemented all basic free Pascal language features including function calls, floating point arithmetic, etc.

Operating Systems: Extending Nach-OS (Unix operating system simulator) — *Autumn 2014*

- Implemented various system calls and process scheduling policies like FCFS, RR, SJF, Unix Scheduler
- Implemented Demand Paging and various page replacement policies like LRU, LRU-CLOCK, FIFO

LANGUAGES/TOOLS/COURSES

Languages (proficient): C, C++, Python, Matlab, Octave | **Languages (Familiar):** PHP, JavaScript, MIPS ISA, Lua

Relevant Courses: Machine Learning Techniques, Learning with Kernels, AI programming, Operating Systems, Data Structures and Algorithms, Advanced Algorithms, Compiler Design, Database Design, Principals of Programming Languages, Probability & Statistics, Linear Algebra, AI techniques in Data Mining, Discrete Mathematics, Abstract Algebra