Vyom Kavishwar | ■ kavishwarvyom@berkeley.edu | **C**

Education	UC Berkeley	
	M. Eng in EECS – Data Science and Systems Recipient of the Fung Excellence Scholarship	Aug 2019 - Present
	B.A in CS, GPA: 3.87	Aug 2015 – May 2019
Internships	Amazon Web Services, Seattle Wrote an internal Python service using AWS API Gateway and Lambda that performs queries, aggregations and analysis over ~20 AWS Elasticsearch clusters	May – Aug 2018
	ClearScore, London Wrote a microservice extraction tool in Scala, using Akka Streams to optimize performance; also worked on a project to run the full tech stack on local dev machines using Docker containers.	May – Jul 2017
Research	Adept Lab (BAIR), Berkeley Working on methods for large-batch distributed training of neural networks, using Hessian-guided optimization and adversarial training	Aug 2019 - Present
	Berkeley Institute of Data Science, Berkeley Developed 3D CNN and ConvLSTM-based models to segment brain MRI images and detect tumors	Oct 2018 – May 2019
Projects	Soccer analysis: Built a tool using OpenCV to track soccer players, estimate their speeds and distances covered, and visualize their defensive line as a convex hull / polygon.	
	Pixel2Mesh: In a group of 4, reimplemented the Pixel2Mesh paper in PyTorch, that converts an RGB image of an object into a 3D Mesh using a mix of regular and graph convolutions.	
	Text summarization: As part of the Data Science Society, used different word/sentence embeddings (Word2Vec, Skip-Thoughts) with K-means clustering to summarize financial reports from the SEC	
	Deep RL: Implemented various reinforcement learning algorithms (Policy gradients, Deep Q-learning, Model Predictive Control) in TensorFlow for a range of Gym / Mujoco environments.	
	Compilers: In a team of 4, wrote a compiler for a statically typed version of Python that converted it to C++ after lexing, parsing, type-checking and optimization.	
Languages / Frameworks	Proficient: Java, Python, Git, Latex, Keras Intermediate: Scala, TensorFlow, PyTorch, AWS, GCP, NumPy, Docker, Elasticsearch, OpenCV	