GAN TU

(626) 628-4911
tugan@berkeley.edu
http://tugan.io

EDUCATION

09.15 — 05.19	**University of California, Berkeley** | *B.A. Computer Science*	Berkeley, CA
	Magna Cum Laude. Department High Honors. GPA: 3.89/4.00	
2016 — 2018	**Udacity Nanodegree Programs** | *Self Driving Cars, Deep Learning, Computer Vision, Digital Marketing*	

EXPERIENCE (DATA SCIENCE)

05.18 — 12.18 **Software Engineering Co-Op,** *Apple Inc,* Cupertino, CA
- Designed and developed innovative machine learning pipelines to detect, extract, and organize event booking details from structured and unstructured information sources (i.e. webpage, text, email).
- Researched, modeled, and tuned NLP neural network architectures to achieve high-precision model ensembles that perform comparably to existing rule-based systems for 9 Latin and non-Latin locales.
- Engineered parallel processing solutions for efficient large-scale information retrieval and data analysis.
- Diagnosed and remodeled the limited, partially labeled provided data source with feature engineering to make models outperform the rule-based system against real-world variations and data skewness.
- Created a web app in Bootstrap and Flask to visualize model behaviors and assist tuning and debugging.
- Prepared detailed technical documentation and unit testing for various parts of the extraction pipeline.
- Devised model improvements for better scalability, faster inference, and lower memory footprint.
- Presented my work to department VP; received recognition for covering ~52% production usage.

09.17 — 05.18 **Undergraduate Researcher,** *Berkeley Artificial Intelligence Research Lab* Berkeley, CA
- Examined decoding performance of deep neural networks on convolutional error correcting codes.
- Researched lightweight, real-time object detection architectures using partial-layer weight quantization.
- Investigated meta-learning for Generative Adversarial Networks via Hierarchical Bayes.

PROJECTS

FULL STACK WEB DEVLOPMENT

NewsWatcher (*MEAN stack: MongoDB, ExpressJS, AngularJS, NodeJS, NewsAPI, Postman, TravisCI, Firebase, AWS, Heroku*)
- Developed a news app that allows authenticated users to subscribe to latest news stories based on user-generated keywords.
- Deployed the website on both Google Firebase and Heroku. Integrated custom Firebase Functions and AWS Lambdas, respectively, to periodically fetch latest news, remove outdated news, filter news by keywords, and populate user home timeline.
- Wrote substantial integration tests for the RESTful server API service, with Postman and TravisCI continuous integration (CI).

MicroPost (*Ruby on Rails, Heroku, PostgreSQL, TravisCI, Redis*)
- Developed a substantial, functional micropost web application with full registration, email activation, and authentication system that allows activated users to post microposts with images, and to follow and unfollow other users for curated home timeline.
- Extended user model to support admin control, giving authorized accounts to delete inappropriate accounts and microposts.
- Integrated comprehensive unit tests and integration tests, with continuous integration (CI) and deployment using TravisCI.
- Drafted out preliminary system design for efficient scaling (e.g. cache using Redis, Read/Write API, different fan-out services).

JellyRobot Controller (*NodeJS, ReactJS, Firebase, ffmpeg*) — *cross-functional team collaboration*
- Developed a simple web interface for receiving live camera streaming from and sending ROS controller commands to a custom-built robotic dog *(developed by another team),* as a MVP prototype for both remote and self-automated robotics control purposes.

SYSTEM & APPLICATIONS

Secure Data Store (*5-month project*)
- Designed and developed a simple *file storage client* that supports file upload, download, sharing, and access revocation.
- Devised and integrated *security solutions* to protect file confidentiality and integrity against various network MITM attacks.
- Enabled *efficient file updates* to sync small changes to large GB-sized files using a custom Merkle tree.
- Programmed the internals of a simply *server side SQL database* with dynamic multilevel indexing using persistent B+ Tree.
- Engineered a Selinger style *SQL query optimizer*, using execution *cost estimation* and various *iterators and join algorithms*.
- Implemented simple database lock manager and transaction logs for *database transactions and concurrency* support.

Graphics App (*3-month project*)
- Implemented a basic *graphic rasterizer* using supersampling for antialiasing.
- Programmed a *mesh editor* that can both render and edit the geometry (vertices, edges, faces) of Collada mesh object files.
- Engineered a *graphic rendering engine,* using path tracing algorithms, for Collada mesh object files. Integrated high resolution rendering support for custom environment maps, shading, lighting, textures, microfacet materials, and depth of field.
- Developed a *mass-spring system simulator* that animates realistic hanging, pinned, falling, colliding, and folding cloth behaviors.
- Improved speed performance of the pipeline by 400x using BVH, Monte Carlo Integration, and statistical sampling methods.

Mini OS (*5-month project*)
- Designed and implemented *the core threading, arbitrary user program execution, and file system* for the Pintos OS framework.
- Designed an *Arithmetic Logical Unit* and a 16-bit two-cycle *processor* for a subset of MIPS instructions set.
- Developed a basic *shell* terminal, a static file *HTTP server*, a *memory allocation* library, and a simple MIPS *assembler* and *linker*.

ChocoPy Compiler (*4-month project*)
- Designed and implemented a *compiler* (from scratch) for the ChocoPy language: *a non-trivial, restricted subset of Python 3.*
- Implemented *lexical analysis, program parsing, semantic analysis, static type checking*, and RISC-V *assembly code generation*.