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

05.17 — 08.17 **ArtCenter College of Design** | *Graphic Design, Product and Transportation Design* Pasadena, CA

2016 — 2018 **Udacity Nanodegree Programs** | *Self Driving Cars, Deep Learning, Computer Vision, Digital Marketing*

2014 — 2017 **Wharton School of Business** | *Certified Online Business Specializations (with Distinction) in 12 Domains*

EXPERIENCE

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.

SKILLS

Languages: Java, Python, C/C++, Ruby, Go, Swift, Kotlin
Front-End: HTML, CSS, JavaScript, Bootstrap, VueJS, ReactJS
Back-End: Ruby on Rails, Flask, AngularJS, PHP, Django
Data Science: PyTorch, Keras, Tensorflow, Scikit-Learn, OpenCV, GraphLab (Turi), SciPy, Pandas, NumPy, MatplotLib
Database: SQL, SQLite, Redis, Postgres, MySQL, MangoDB
Virtualization: VM Ware, VirtualBox
Containerization: AWS ECS, Docker
Continuous Integration: Travis CI, Circle CI

Built Tools: Shell Scripts, Make, Maven, Rake, Grunt, Gulp
Unit Testing Frameworks: Unittest, JUnit, Minitest, RSpec
Cloud Infrastructure: AWS, Google Cloud, Azure
Graphics: OpenGL, ThreeJS
UI/UX Design: Sketch App, Avocode, Adobe XD, etc.
Office Suite: Microsoft Word, Excel, Powerpoint, Pages, Numbers, Keynote, Slack, Adobe Acrobat, LaTex, etc.
Creative Design: Adobe Photoshop, Lightroom, Illustrator, Dimensions, Final Cut Pro, iMovie, Autodesk Maya, Unity

PROJECTS

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*.

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.

ChocoPy Compiler *(4-month project)*

- Designed and implemented a *compiler* (from scratch) for the ChocoPy language: *a non-trivial, restricted subset of Python 3 with Python 3.6 type annotations to enforce static typing, specified using formal grammar, typing rules, and operational semantics.*
- Implemented *lexical analysis, program parsing, semantic analysis, static type checking*, and RISC-V *assembly code generation*.

Graphics Project *(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 statical sampling methods.