

CHRISTOPHER DINH

Irvine, California 92612 | (240) 535-3571 | cqdinhh153@gmail.com | [linkedin.com/in/cqdinhh](https://www.linkedin.com/in/cqdinhh) | github.com/cqdinhh | cqdinhh.github.io

EDUCATION

UNIVERSITY OF CALIFORNIA, IRVINE, Irvine, California

Master of Computer Science, Expected December 2020, GPA 3.97 / 4.0

UNIVERSITY OF MARYLAND – BALTIMORE COUNTY, Catonsville, Maryland

B.S., Computer Science - Data Science Track, 2015-2019, Major GPA: 4.0 / 4.0

B.S., Mathematics, 2015-2019, Major GPA: 3.9 / 4.0

magna cum laude, Cumulative GPA 3.8 / 4.0

TECHNICAL SKILLS

Languages:	Python (Expert), C++ (Proficient), C(Proficient), Java (Proficient), C# (Prior Experience)
Web Development:	Javascript, HTML / CSS, node.js
Libraries:	PyTorch, Tensorflow / Keras, OpenCV, Scikit-learn, Pandas, NumPy
Other:	MySQL, Jupyter, Git, Visual Studio Code, GDB, Linux / Unix, Google Colab
Deployment:	Google Cloud Compute, Streamlit.io

EXPERIENCE

SWAPPIT CO, Irvine, California

ML Volunteer Project, 6/2020 – 9/2020

- Designed and implemented a deep learning model to predict the aftermarket prices of sneakers based on real data
- Initial model prototyping with Scikit-learn and PyTorch and final implementation in Tensorflow / Keras
- Given 60k training instances, the model's predictions average within \$56 of the true sale price

IBM, Rochester, Minnesota

Software Engineering Intern, Cloud Managed Application Systems, 6/2018 – 8/2018

- Enabled SAP build teams to track their progress and find bottlenecks by designing and implementing a Gantt chart visualization of the SAP build process using node.js, Javascript, d3.js, PUG, and CSS.
- Learned and practiced Agile software development practices

UMBC MULTI-AGENT PLANNING AND LEARNING LAB, Catonsville, Maryland

Undergraduate Researcher, 9/2017 – 5/2018

- Used Java to upgrade an AI hierarchical planning system to avoid unnecessary actions by adjusting to the environment in real time. [Code](#)

MANTAROBOT CORP, Germantown, Maryland

Summer Intern, 5/2014 – 8/2014, 5/2015 – 8/2015, 5/2016 – 8/2016, 5/2017 – 8/2017

- Designed and implemented an OpenCV-based system in C# that uses a camera to automatically dock a robot with its charging station with its dock from up to 7 feet away with an angular error of less than 10 degrees and a linear error of less than 3 inches.
- Minimized stress on the joints of a robotic arm by finding the orientation that moves the end of the arm to a specified position with the lowest possible torque. Python was used for prototyping and simulation before the final system was implemented in C#.
- Decreased control latency for a telepresence robot by an average of 50% by implementing WebRTC as a control method both in the browser and in an Android app.

PROJECTS

Toxic Comment Classification • *Deep Learning Course Project* • [Code](#) • [Kaggle](#)

Deep Learning model built in PyTorch that uses BERT to determine if online comments are toxic

- Built a custom head on top of BERT that uses an attention mechanism to process comments longer than BERT's 512-token limit. After fine-tuning, the model achieved an average AUC of 0.959 across 6 types of toxicity.
- Trained and deployed the model on a Google Cloud Compute instance using a custom model server that is accessed by a Streamlit.io app.

Capsule Network vs Convolutional Network • *Computer Vision Course Project* • [Code](#)

Showing that Capsule Networks are more robust to homographies than Convolutional Networks

- Found that Geoff Hinton's CapsNet architecture is more robust to homographies than a similarly-constructed CNN by implementing both in GPU-accelerated PyTorch.
- Reached 73% classification accuracy with the CapsNet while the CNN only reached 18% despite the ConvNet using 10x more parameters than the CapsNet.

Inferring Ingredient Relationships from Recipes • *Natural Language Processing Course Project* • [Code](#)

Embeddings for ingredients learned based on a dataset of recipes

- Generated four vector representations of ingredients based on a recipe dataset using co-occurrence matrices, PCA and a modified Continuous Bag-of-words model.

CycleGAN Implementation • *Personal Project* • [Code](#)

GAN architecture for bidirectional transformation between two sets of images

- Used GPU-accelerated PyTorch to implement the CycleGAN architecture which learns to transform images between two image sets. In the original paper, the authors trained the architecture to transform pictures of horses into pictures of zebras and vice versa.

Corporación Favorita Grocery Sales Forecasting • *Data Science Course Project* • [Code](#) • [Kaggle](#)

Kaggle competition to predict sales for a large Ecuadorian grocery chain

- Cleaned and reshaped data and trained a multi-layer perceptron on it despite significant memory limitations relative to the size of the dataset. After training for 8 hours, my model reached a Normalized Weighted Root Mean Squared Logarithmic Error of 1.062.

Sequence Generator • *Personal Project* • [Code](#) • [Demo](#)

Text generator using an LSTM to mimic a text corpus

- Built a Tensorflow-based LSTM architecture based on Andrej Karpathy's char-rnn that generates arbitrary-length text mimicking whatever text corpus it is trained on.
- Built a demo in Javascript from scratch that uses a pretrained model to generate text based on the works of Shakespeare

Health Professional Time Scheduling • *Software Engineering Course Project* • [Code](#)

Desktop application to automatically assign shifts to nurses and doctors at Johns Hopkins Bayview Pediatric Emergency Department

- Worked in a team of 5 using Agile development practices and UML to design a desktop application that automatically generates a shift schedule for doctors and nurses subject to various constraints. The Frontend GUI is built in Electron and Vue and when generating the schedule, it calls a Java backend that uses Optaplanner to generate the schedule.

Google CodeU • *Personal Project* • [Code](#)

Invite-only program working on a team with 2 other students supervised by a Google engineer

- Implemented features for a chat application in Java including multithreaded update polling and storage of server state in a SQLite database through JDBC.
- Participated in regular code reviews with a Google engineer and learned industry practices including unit testing, trunk-based development, collaboration using Github, and uniform code style.