

# Dennis (Taeyang) Kim

647-913-0934  [taeyang.kim@mail.utoronto.ca](mailto:taeyang.kim@mail.utoronto.ca)  [tydennis.me](http://tydennis.me)  [alvolate](https://alvolate.com)

Toronto Ontario

## Education

---

### University of Toronto

2021-2024

Honours Bachelor of Science, Computer Science, Co-op

Relevant Courses: Software Tools & Systems Programming, Machine Learning and Data Mining

International Student Ambassador

## Skills

---

**Languages:** Python, Java, C, JavaScript, Bash

**Technology/Frameworks:** ReactJS, Git, OpenCV, NodeJS, MySQL, Pytorch, JUnit, Mockito

## Experience

---

### Korea Advanced Institute of Science and Technology

Aug 2020- Nov 2020

Research Intern

- Trained a convolutional neural network, working heavily with **PyTorch** and **Anaconda**, allowing the model to classify images with accuracy with 86.7%
- Extended the model using **OpenCV** to enable real-time face recognition for autonomous car test
- Responsible for the data plotting feature, an important component of the project that enabled the analysis of the effect of epochs in machine learning
- Participated weekly meetings to discuss challenges and analysis, which facilitated faster development time

## Projects

---

### Course Offering Calendar

August 2022

- Used **Gradle** and **Java** to create a course calendar that considers prerequisites and session offerings
- Implemented Real-time **Firebase** to enable real-time updates of user-generated data for all other users of the application
- Utilized **Mockito** to facilitate unit testing, which allowed for fast development of the program and improved code quality
- Conducted **Scrum** meetings as scrum master and used **Jira** to keep track of the user stories to allow better communication and collaboration

### Movie Web Service

August 2021

- Designed and built a website to fit user specifications
- Employed **React** and **Axios** to build a fast, static website that could easily synchronize data with the server
- Utilized **JavaScript** for the front-end to display movie details obtained from a .json query