# Jierui Peng

Email: Jerrypeng999@gmail.com | Tel: (781) 530-6298 | https://github.com/JerryPeng0201

## EDUCATION BACKGROUND

**Brandeis University, MA** 09/2017 – 05/2021
B.S. in Computer Science & Economics

**Core Courses:** Multivariable Calculus | Linear Algebra | Probability and Statistics | Discrete Structures | Data Structures and the Fundamentals of Computing | Programming in Java | Advanced Programming (Java) | Operating System | Computer Graphics | Digital Photography | Spoken Dialog Design | Web Application Development | 3D Animation | Artificial Intelligence | Practical Machine Learning with Big Data | Google IT Support

## STANDARDIZED TESTS

**GRE General:** V: 160 / Q: 170 / Total: 330

## SKILLS

**Programming & Software:** C/C++ | Java | JavaScript | Python | HTML | CSS | MySQL | PostgreSQL | MongoDB
**Libraries:** NumPy | SciPy | Pandas | Scikit-Learn | TensorFlow | Scrapy | Matplotlib | Seaborn | Selenium | dplyr | ggplot2 | shiny
**Frameworks:** Spring | Spring Boot | MyBatis | Node.js | Angular 6 | Express.js | Flask | jQuery

## PUBLICATIONS

- **Peng, J.**, Long, Y., Peng, R., Hu, Z. (2020) *Novel Fast Corner Detection Descriptor in Image Processing*, International Core Journal of Engineering, In Press
- Lin, J., **Peng, J**., Hu, Z., Xie, X., Peng, R. (2020) *ORB-SLAM, IMU and Wheel Odometry Fusion for Indoor Mobile Robot Localization and Navigation*, Academic Journal of Computing & Information Science, 3(1), 131-141

## EXTRACURRICULAR EXPERIENCES

**Teaching Assistant | Brandeis University** 09/2018 – 12/2018

- Assisted the instructors in designing and preparing course materials for Java Programming and Data Structures courses.
- Offered weekly tutorial to 300+ undergraduate students; held Q&A sessions on a weekly basis; graded assignments.
- Guided students to complete team-oriented software building projects.

## WORK EXPERIENCES

**Software Engineer Intern | Google LLC** 06/2020 - 08/2020

- Assisted the creation of a universal ML pipeline that handles the typical dataset available in the market, which includes preprocessing the data, transform the preprocessed data, and create the model, then serve it as an API.
  - Preprocess CSV dataset to infer data types and do predefined feature transformation accordingly.
  - Tuning model with Keras Tuner to find the best Hyperparameter combination of the neural network
  - Train the model with a subset of the dataset to get the best-performed model
  - Deploy the Keras model to Flask local server as a REST API, and make predictions with it.

**Software Engineer Intern | KALO Inc., New York** 07/2019 - 08/2019

- Built an integrated data retrieval system in Python and JavaScript for handling static and streaming data on a social shopping e-business platform, focusing on scalability, durability, robustness, and computational efficiency; streamlined the data retrieval pipeline, allowing 30% increase in speed.
- Leveraged an array of machine learning algorithms (e.g., deep learning, decision tree, SVM) to upgrade the product search & recommendation module on the platform, leading to greatly improved customer experiences.

## RESEARCH EXPERIENCES

**Brain Disease Diagnosis Based on Medical Image Analysis Using Deep Learning Algorithms**
**Research Assistant, UC Berkeley, CA | Advisor: Prof. Noah Gift** 04/2020 – 05/2020

- Gained familiarity with Linux programming (Ubuntu), CUDA parallel computing, Python programming in Anaconda, and various Python machine learning libraries, e.g., PyTorch, Kera, and TensorFlow.
- Built, trained, validated and tested an array of machine learning models for analyzing brain MRI features, allowing brain disease diagnosis, brain tumor segmentation, survival prediction, and the detection of electro-encephalic anesthesia:
  - Evaluated the performance of a novel deep neural network structure, namely Feature Mining Networks (FMNet), for brain tumor segmentation, covering semantic information mining unit (SIMU), macro information mining unit (MIMU) and feature correction unit (FCU).
  - Optimized the PAU-Net framework by shortening the distance between output layers and deep features; applied bottom-up path aggregation (PA) encoder for noise reduction and information retention.

**ORB-SLAM, IMU and Wheel Odometry Fusion for Indoor Mobile Robot Localization and Navigation**
**Research Assistant, The Hong Kong Polytech University, Hong Kong** 01/2019 – 03/2019

- Developed a light-weight multi-sensor fusion method for the localization and navigation of an indoor mobile robot in a GPS-denied environment based on 1) ORB-SLAM, a keyframe and feature-based monocular simultaneous localization and mapping (SLAM) algorithm, 2) an inertial measurement unit (IMU), and 3) wheel odometry.
- Leveraged the angular data of IMU and linear data of wheel odometry to enable the localization of robot so as to support the ORB-SLAM calculation.
- Modified the monocular camera pose of ORB-SLAM to obtain global robot pose estimation via mapping scale iteration.

**Novel Fast Corner Detection Descriptor in Image Processing**
**Research Assistant, The Hong Kong Polytech University, Hong Kong** 04/2019 – 06/2019

- Presented a novel fast corner detection method and performed experimental validation for several images

- Extracted and marked all the corner pixels via iterating each pixel of the image and applied to Fast Corner Detection algorithm that proved to time-efficiency for image data processing.
- Designed the algorithm with the following methodology:
  - defined a threshold k. Calculate the pixel difference between P1, P9, P5, P13 and center P (m × n × 3 data matrix of an image). if at least three of their absolute values exceed the threshold value, they will be considered as candidate corner points for further investigation; otherwise, they cannot be corner points.
  - If P is a candidate point, the pixel difference between the 16 points P1 to p16 and the center P is calculated. If they have at least 9 consecutive points exceeding the threshold value, they are corner points; otherwise, they cannot be corner points.
  - for non-maximum suppression of image, if there are multiple feature points, then the s value of each feature point is judged; If P is the largest response value of all feature points in the neighborhood, then keep it; if not, then abort the judgement; If there is only one feature point (corner point) in the neighborhood, it is reserved.

**Parkinson Detection with OpenCV and AutoML, Brandeis University** 04/2020 – 06/2020

- Collectively leveraged Google AutoML and the Histograms of Oriented Gradients (HOG) for Human Detection algorithm to build a software application for analyzing a hand-drawn spiral diagram so as to allow early diagnosis of Parkinson syndrome.
- Deployed the program on Android and iOS mobile devices, allowing easy access and user-friendly control.

***JBS Incubator & Web Application Development*** **Course Projects, Brandeis University** 06/2018 - 08/2018

- Led a team of five to develop an integrated college information system, allowing 1) course search and scheduling, and 2) online flea market services; key components included:
  - A chat box allowing users to create, query and delete daily schedules using text or voice.
  - A voice control system based on Google's DialogFlow, allowing voice dialing, call routing, domestic appliance control, keywords search, simple data entry, and speech-to-text processing.
  - A user-friendly frontend based on HTML, CSS and JavaScript.
  - Applied Async parallel steps to optimize system by reducing the negative interruption in data order and nesting of callbacks.
  - Used Mango DB toolkit as the database to store courses and classrooms information and scheduling information of students. Introduced AJAX framework to realize the connection to the database to reduce loading and responding latency.
  - Product recommendation based on collaborative filtering and content-based filtering.
- Improved the identification algorithm based on the users' voice instead of using self-input information, followed various software engineering principles and design patterns to streamline development processes, focusing on decent efficiency, modularity, extensibility, maintainability and robustness.
- Organized brainstorming, alignment, and trouble-shooting meetings to drive development process; significantly bolstered leadership skill.

## COURSE PROJECTS

***Computer Graphics*** **Course Project, Brandeis University** 04/2018

- Implemented in C++ 1) Phong Local Illumination in OpenGL, 2) a shading algorithm, 3) a light reflection algorithm, 4) a diffuse reflection generator based on recursive ray tracing, and 5) a super-sampling algorithm for reducing aliasing effect.
- Developed a software application in OpenGL and C++ for generating 3D model with volume rendering based on a stack of 2D CT medical scan images:
  - Built algorithms to calculate the normal vector and opacity of the mesh during the 3D model reconstruction.
  - Designed and implemented a "scaffold" mesh data structure to drive the 3D model reconstruction.
  - Developed a program to generate position matrices based on the input image sequence.

***Artificial Intelligence*** **Course Projects, Brandeis University** 01/2019 - 05/2019

- Systematically studied Markov decision processes, reinforcement learning, decision trees, HMM, particle filtering, game tree, etc, and their applications to probabilistic robotics and robotic inference.
- **Project I:** Implemented in Python multiple pathfinding algorithms, covering A*, Dijkstra's, DFS, BFS, Expectimax, and genetic algorithm.
- **Project II:** Built, trained, validated and tested a deep learning model in Python to create poems by 1) using Markov Chains to build new sentences, and 2) the LSTM Recurrent Neural Network (RNN) to predict the properties of the next line of the poem, covering an array of syllables and rhyme schemes.

***Practical Machine Learning with Big Data*** **Course Projects, Brandeis University** 01/2019 – 05/2019

- Built, trained, and validated a LSTM CNN model to enable earthquake prediction and early-warning; collected five decades of American earthquake data to support model building, covering locations and magnitudes.
- Built a mobile application in TensorFlow Lite and OpenCV for detecting door number in video frames, attaining an accuracy of 95%.

***Advanced Programming in Java*** **Course Projects, Brandeis University** 01/2018 - 05/2018

- Designed and implemented in Java a flight video game named "Thunder Fighter", covering an array of weapons, level layouts, and layered visuals:
  - Collectively used JFrame, HashMap, and double buffering to create desired visual effect and computational efficiency.
  - Performed multithreading programming in Java, focusing on synchronization and inter-thread communication
  - Leveraged an array of object-oriented programming features in Java and fit-for-purpose design patterns to improve the modularity and robustness of the application.
  - Optimized the game design to enhance the player experience, covering mechanics, feedback, pacing and interface.