

# BENJAMIN MACCINI

[benjaminmaccini@gmail.com](mailto:benjaminmaccini@gmail.com) • (512) 705-3797 • [benjaminmaccini.info](mailto:benjaminmaccini.info) • Austin, TX, United States

Developer with focuses in early-stage product development, algorithms, devops, and information security.

## EXPERIENCE

---

### **Pensa Systems, Algorithm Engineer**

**Jun 2019 – Jun 2020**

- Led design and execution of cross-team projects examples being;
  - Diagnostic tooling for runtime services utilizing the REST protocol
  - Cloud-based microservices for image preprocessing
- Evaluation and QA of deep learning models
- Development and maintenance of supporting infrastructure for various steps of the machine learning life-cycle
- Modernization and optimization of proprietary localization algorithms

### **Pensa Systems, Intern/Contractor**

**Feb 2018 – Jun 2019**

- Led independent contractor coordination, internationally, namely for the purpose of data collection and labelling
- Engaged in field operations for experimental trials
- Responsible for evaluating model accuracy measures

### **JRS International, Window Cleaner**

**Mar 2017 – Dec 2017**

Contract work specializing in high-rise window cleaning services

## EDUCATION

---

### **University of Texas at Austin**

- B.S. in Mathematics (College of Natural Sciences)
- Minor in Business (McCombs School of Business)
- Elements of Computer Science Certificate (College of Natural Sciences)

### **Relevant Coursework/ Independent Pursuits**

Applied number theory, applied linear algebra, scientific computation in numerical analysis, elliptic curve cryptography, stochastic analysis, elements of statistical learning, one-shot learning

## SKILLS

---

### **Languages // Packages**

- Python
  - Packages: Flask, SQLAlchemy, numpy, scipy, opencv, pandas, tensorflow, boto3
- Go
  - Packages: Viper, Hugo (static site generator), gin, go-centrifuge, gonum

### **General Software Methodologies/Patterns/Tools**

- REST
- Asynchronous programming
- Microservice architecture
- Cloud native applications (AWS: Lambda, ECS, SQS, S3, Cloudwatch)
- Test Driven Development
- Readable Code
- Continuous Integration (CircleCI)
- DevOps (terraform)
- “AGILIE” Methodologies
- Git
- Unix development
- Team coordination tools (Google Services, Kanban, Slack)