Xinyue Zhang

☎ (+86)15695217921 | ✉ 213171730@seu.edu.cn | verBubble

Education

Southeast University *Nanjing, China*

Bachelor of Electronic Science and Engineering *2017.09–2021(expected)*

- **Imperial College London** London, Britain — *Data Science Summer School from July to August in 2019*

Skills

Languages	Python, C, C++, matlab
Frameworks	PyTorch
Libraries	Opencv-python, PyQT, QT
Others	Basic knowledge in data structures and algorithm.

Competition and Project Experience

SRTP(Student Research Training Program) Project in National Level *Nov. 2018 - present*

Team leader

We deploy style transfer algorithm in embedded devices called Jetson nano and show real-time results in Glasses. The original implementation of style transfer cannot run fast enough in Jetson nano, so we use tricks like width multiplier and cuda to minimize model's size and speed fps(frame per second) up from 4 to 21. Also, we have one software copyright for Style Transfer Platform and one patent application has beed submitted. Furthermore, I deploy an interface for real-time style transfer[link]. Styles can be changed by pushing buttons. Code for style transfer with width multiplier can be found [here].

Zhejiang LAB CUP GLOBAL AI COMPETITION MOT track *Aug. 2019 - Sep. 2019*

No. 9 out of total 234 participates

I participate in the 2019 Zhejiang LAB CUP GLOBAL AI COMPETITION MOT(Multiple Object Tracking) track. By employing the cascade mask-rcnn to detect the Pedestrians and a deepsort algorithm to associate object across different frames, We achieves No. 9 out of total 234 participates.

Southeast University's 13th smart car competition *Nov. 2018 - Mar. 2019*

Third Prize

We use PID(proportion, integral and differential) algorithm to control Electromagnetic smart car. After implementing algorithm in C, we fine-tune each parameter to achieve better results.

A C++ implementation of software renderer and ray tracing.

I wrote a basic renderer to understand how OpenGL works, including object modeling, geometry transformation, lighting model, rasterization and texture mapping. Also, I wrote a tiny ray tracer to know basic ideas about rendering. Code can be found [here].