Yong Wei

Linkedin

Email : weiyong1024@gmail.com
Mobile : (+86) 186-1263-2313

Education

- **Beihang University** Beijing, China
  Master of Engineering in Electronic and Communications *Sep. 2016 – Jan. 2019*
- **Beihang University (Honors College)** Beijing, China
  Bachelor of Engineering in Electrical and Electronics; GPA: 3.7 *Sept. 2012 – Jun. 2016*

Experience

- **Pony.ai** Beijing, China
  Software Engineer - Infrastructure *Feb 2019 - Mar 2020*
  - **Voice Logging Pipeline**: Built the pipeline of voice logging feature, allowing operators without computer-science background to record necessary info by voice. For onboard part, first chose relatively optimal devices with hardware team and wrote drivers for them. Then added a demon process to manage the trigger and termination of recording events. For offline part, using Speech-To-Text service by Google Cloud to extract the content of recorded sounds, serializing original audio files into Google Protobuf and integrated meta info with the current Issue Reporting pipeline.
  - **Car Sound Workflow**: Updated Car Sound system supporting I18N, imporving the HMI for operators and passengers outside English speaking countries. First, built an internal tool to manage car sound files and suite with three main functions: Generate voice files by calling AWS Text-To-Speech service, download from and upload to Pony storage service, and version control. Then integrate the car sound playing feature with current onboard pub-sub system.
- **Airbnb** Beijing, China
  Software Engineer Intern *Jun 2018 - Sep 2018*
  - **Host Retrospect Page**: Retrospect page for Airbnb host users. Added related endpoints and mobile web pages on Ruby On Rails framework. Discussed with the designers on the contents and formats of web pages.
- **Megvii (Face++)** Beijing, China
  Research Intern *Jan 2018 - Jun 2018*
  - **Model Search**: During the first half, after the dimension of the input graph increased by 14%, we aimed to reduce the FLOPs (float operations per second) of the CNN (convolutional neural network) of face recognition module on mobile devices to the same level as before, without damaging the performance of the whole model which is measured by $\frac{1}{10,000}$ passing rate (ROC value when $x = \frac{1}{10,000}$). The problem was solved by adding a bottleneck layer before the Inception-ResNet module to compress channels of the input feature graph by $\frac{1}{2}$. As for the second half, we aimed to improve the performance of our face recognition module referring to state-of-the-art CNN architectures. After experiments on Xception, DenseNet and several other architectures, I proposed a modified version of Google Inception V4 for our production: Instead of using the same atom modules everywhere, different Inception-ResNet modules were used for the shallow, medium and deep part of the CNN. On the other hand, the NbyN kernels of the medium and deep part were replaced by a sequence of $1 \times N$ and $N \times 1$ kernels to improve model capacity. As a result, the $\frac{1}{10,000}$ passing rate was improved by 1% on most benchmarks.
  - **Data Cleaning**: Made a bunch of original multiracial face data (including $120,000$ pictures and $6,000$ videos) into benchmarks on Linux server by writing Python scripts.
- **Beihang University** Beijing China
  Research *Jul 2015 - Mar 2016*
  - **Research Assistant - The 5^{th} Generation Mobile Network**: Proposed an channel estimation method based on uplink wireless data and channel sparcity with supervisor, improving upper bound of the wireless system's throughput by 28% according to simulation results.
    Paper published in Journal of Signal Processing (First author). DOI: 10.16798/j.issn.1003-0530.2017.06.002

Competitive programming

- **Google Code Jam Kickstart 2017 Round F**
  Top 5%, rank 108^{th} globally. ***scoreboard***

Programming Skills

- **Languages**: C++, Python | Shell, JavaScript, **Technologies**: AWS, Google Cloud | Kubernetes, React