ONNX & Cognitive Service Read API

Yung-Shin Lin
Microsoft Azure/Cognitive Service/Text Intelligence
OCR Core Engine Team
Principal Engineering Manager

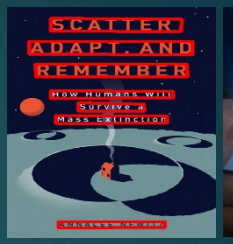
Microsoft Cognitive Service Read API

- Microsoft Cognitive Service
  - Cloud-based AI offering.
- Our team: Read API
  - A state-of-art Optical Character Recognition (OCR) engine using deep learning models.
- Read API Features:
  - Support both printed text and handwritten text.
  - Great for real-world images.
- Search for “Cognitive Service Computer Vision” for more info.

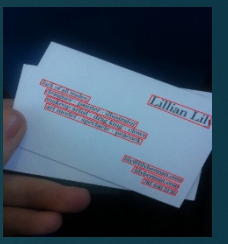
Street View

Product Label

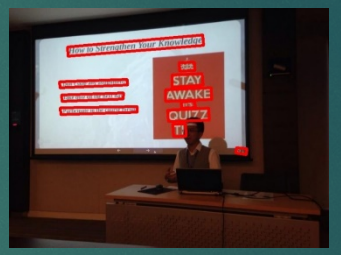
Poster



Business Card



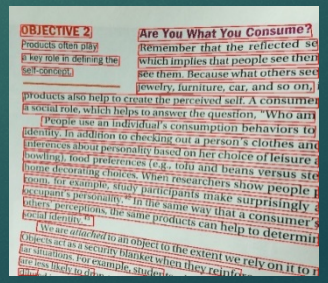
Slide



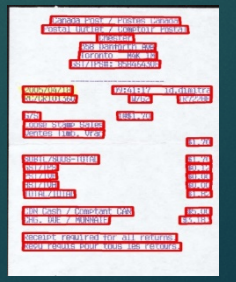
GIF

Book Cover

Document



Receipt



Invoice

Challenges before Using ONNX

- In late 2018, need a modern training and inference framework.
  - Previously, we used a home-grown inference engine.
  - Researchers was switching to PyTorch to do research work.
  - Mismatch between research and production.
- Performance challenges
  - Our home-grown inference engine needs to achieve at least 3X speed boost to reduce operation cost in running service.
  - Caffe2 did not meet our performance expectation too.

Why Switching to ONNX?

- A modern framework for research, training and inference.
  - PyTorch for researching and training
  - ONNX runtime for inference
- Straight from research to production
  - Researcher's PyTorch model can export ONNX model directly
  - Easy to port researchers' Python inference code to our production C++ code.
- Runtime Performance Boost
  - ONNX runtime is highly optimized for interference.
  - Runtime is well suited for service deployment.
  - Great support from ONNX runtime team.

Read 2.0 Porting Experience

- Read 2.0 is a Windows-based cloud service
- Use Caffe2 to export our models to ONNX
- Use Custom Ops support in ONNX to extend op support
- Use ONNX C++ API in Windows
  - Very Easy to use
  - Take about 2 weeks to port and test.
- Focus on verification and performance tuning
- Service deployment testing

Performance Gain via ONNX Runtime

- 3.7x latency improvement compared to our home-brew inference framework.
- Helped us to over-achieve our 3X perf goal.

Read 3.0 Preview: Road to Linux

- For Read 3.0 preview, we wanted to move Kubernetes-based service
  - Better scalability and reliability.
- Business demands on Linux containers.
- We need to port from Windows to Linux.
  - The cross-platform nature of ONNX runtime helps us to migrate to Linux very easily.
- Even faster inference in Linux.

Takeaway for our ONNX Journey

- ONNX
  - Great from research to production.
- ONNX Runtime
  - Great for easy-of-development, deployment and performance.
  - Great fit for Kubernetes deployment and container offer.
- Our new feature development now centralized in ONNX and ONNX runtime.