Contribution ID: **38** Type: **Oral Presentation**

Machine Learning Infrastructure on the Frontier of Virtual Unwrapping

Tuesday, 23 March 2021 15:40 (20 minutes)

Virtual unwrapping is a software pipeline for the noninvasive recovery of texts inside damaged manuscripts or scrolls via the analysis of three dimensional tomographic data, typically X-ray micro-CT. Recent advancements to the virtual unwrapping pipeline include the use of trained models to perform the "texturing" phase, where the content written upon a surface is extracted from the 3D volume and projected onto a surface mesh representing that page. Trained models are critical for their ability to discern subtle changes that indicate the presence or absence of writing at a given point on the surface.

The unique datasets and computational pipeline required to train and make use of these models make it a challenge to develop succinct, reliable, and reproducible research infrastructure. This paper presents our response to that challenge, and outlines our framework designed to support the ongoing development of machine learning models to advance the capability of virtual unwrapping.

This framework is designed around a set of principles chosen to reflect our team's needs from this research area. We emphasize the following as the primary design principles, and briefly discuss our approach to each: 1. Visualization: Direct visualization of a trained model's ability to reveal text from inside a scanned artifact is the most important tool we have to evaluate the model performance. The result of applying a trained model to a mesh inside a volume should ideally be a clear image revealing the cultural object's content. The nature of the datasets requires that these images are created by our own custom Python scripts rather than image operations built into existing frameworks. For example, we generate training videos for each job to show how the output improves throughout training. We also often split an image into k regions for k-fold cross validation during training and make use of SLURM job arrays to perform these in parallel.

1. Automation: These complex custom pipelines need to be executed across hardware environments, often training in the University compute cluster on GPUs and then being used on desktop machines. We make use of Singularity containers to ensure a reliable computing environment. SLURM job dependencies enable summary and results upload jobs to be executed after the primary training job(s) has completed.
2. Data: We work with large datasets that are also frequently updated, and generate many output files for each experiment run. Google Drive is used to store both the datasets and results, and rclone is used to perform automated transfers ensuring job results are immediately visible to all team members.
3. Metadata: Our own methods and code are constantly changing, so we have to keep careful metadata to ensure our experiment results are clear to others or to ourselves in the future. All information required to reproduce any of our experiments is automatically saved to a JSON file which is uploaded upon job termination along with the job results.
4. Benchmarks: There are no existing performance benchmarks for these models, so it has been critical to develop our own benchmark datasets to compare our model performance across different methods and implementations.

We discuss the research improvements enabled by this framework, and in-progress and future improvements to the pipeline. We additionally discuss how this pipeline has prepared our team for performing active research in the midst of the unique challenges posed to the global research community in the past year. Finally, we discuss the ways in which this framework can be applied to a variety of other research domains.

Primary authors: Mr PIKE, Charles (University of Kentucky); CHAPPELL, Jacob (University of Kenucky); PARSONS, Stephen (University of Kentucky); Prof. SEALES, William B. (University of Kentucky)

Presenter: PARSONS, Stephen (University of Kentucky)

Session Classification: Humanities, Arts & Social Sciences Session

Track Classification: Humanities, Arts, and Social Sciences (HASS) Applications