J. Willard Marriott Library
THE UNIVERSITY OF UTAH
295 South 1500 East Salt Lake City, Utah 84112-0860 (801) 581-8558

Machine learning meets library archives: Image Analysis to generate descriptive metadata

(Work on this project will continue under a new title - “Sheeko: A computational helper to describe digital images”)

Harish Maringanti
Associate Dean for IT & Digital Library Services

Dhanushka Samarakoon
Assistant Head, Software Development

Bohan Zhu
Software Developer

This project was made possible in part by a 2018 award from the Catalyst Fund at LYRASIS.

Sheeko

I. Goal

Primary goal of our project titled “Machine learning meets library archives: Image Analysis to generate descriptive metadata” that was generously funded in part by the Catalyst Fund at Lyrasis, was to test the feasibility of applying machine learning techniques to extract useful information (metadata) from images that will assist metadata experts.

As cultural heritage organizations digitize more and more cultural and historical treasures in the hopes of making them available to the general public, there is a growing need to – (1) address the time lag between acquiring these materials to when they become available for public access; and (2) improve the discovery experience for users so they can find relevant information quickly. Though several factors contribute to this perceived time lag issue, one key issue that seems to be more prominent is the limited availability of trained professionals who can create metadata, which is the central component of modern digital library systems. Without quality metadata, it would be impossible to discover relevant data in the age of information overload. Our project aims to address this by applying the latest machine learning techniques on library image collections and generating useful keyword/caption recommendations for the metadata experts.

II. Model/Process

Main part of our work involved training and retraining machine learning models on existing digital library data. Below is a brief description of various steps involved in the process:

a. **Technical Platform setup**: This step involved setting up the necessary hardware equipment and selecting the machine learning platform that would be appropriate to this project. We chose TensorFlow for our project because of ease of use and availability of pre-trained models that could output captions and keywords. We used the following hardware configuration to develop and train models.

   Hardware Configuration:
   2x Xeon 6 core
   32GB ECC Memory (4x8GB minimum configuration supported in 2-socket config)
   1600W Redundant PSU
   10Gb networking
   GPU capacity: 3 double-width GPUs
   Nvidia GForce RTX 2080 Ti
   Ubuntu 18 Operating System
   Attached network storage (Qumulo) for file storage

   In addition, the following code package dependencies were used.
   1. NLTK
   2. Numpy
   3. punkt

4. Tensorflow (version >= 1.12.0)
5. Bazel
6. Cython
7. contextlib2
8. matplotlib
9. spaCy
10. python-resize-image
11. protobuf
12. CUDA Toolkit

b. Data Preparation: The next step involved data preparation. We currently host approximately 450,000+ manually curated historical images in our digital library. The existing metadata (including keywords and captions) was created by subject experts and is of very high-quality. We developed a local script that accepts metadata records in JSON format and the associated image files. The images and metadata fields for each collection would be placed into its own directory. The script will treat each directory as a single collection. One can define the percentages of data that should be used in the training process within this script. In our instance, 80% of images from each collection were used in the training process, and remaining 20% were used for testing. The script then processes the images and the metadata and generates TF (TensorFlow) records that can be used in the training process.

c. Training Model: After preparing the data, we started on training existing models using that data. There are multiple available models that generates captions and labels for digital images (https://github.com/tensorflow/models). Within the duration of this project, multiple models were trained with both external and internal data sources, and more information about these models is below:

Im2txt Caption generation model: "The Show and Tell model is a deep neural network that learns how to describe the content of images"[1]. This code packages the encoder, which does the image classification, and the decoder that is the natural language constructor. This makes use of the models listed below for its operation. The original training code was a combination of python and bazel. This code was locally modified so it could be run and supported using only python for better portability. The original inference code was enhanced locally for multi-model support and to output the annotations in a JSON format.

- MSCOCO base model: This model starts with Inception V3 checkpoint, and trained with 3 million steps on MSCOCO data.
- MLib Model A: Inception V3 checkpoint trained with 1 million steps on Marriott Library Digital Collection photographs with original captions
- MLib Model B: Inception V3 checkpoint trained with 1 million steps on Marriott Library Digital Collection photographs with captions modified using NLP (Natural Language Processing)

[1] https://github.com/tensorflow/models/tree/master/research/im2txt

- MLib Model C: MSCOCO base model trained with 1 million steps on Marriott Library Digital Collection photographs with captions modified using NLP

Image Classifier (Single Label Generator): This code package classifies the images into a class (label) from a given list of a finite set of classes (labels). The original code was enhanced locally for multi-model support and to output the annotations in a JSON format.

- Inception V3 checkpoint: This model is trained for the ImageNet Large Visual Recognition Challenge using the data from 2012. This is a standard task in computer vision, where models try to classify entire images into 1000 classes from WordNet 3.0.

Object Detection (Multi-Label Generator): This is a code package that classifies the images into a class (label) from a given list of a finite set of classes (labels). The original code was enhanced locally for multi-model support and to output the annotations in a JSON format.

- Faster_rcnn_inception_resnet_v2_atrous_oid: This model is trained on Open Images Dataset V4 using 545 classes. On average, this performs at 727ms for inference per image.
- Faster_rcnn_inception_resnet_v2_atrous_oid_v4: This model is trained on Open Images Dataset V4 using 545 classes. On average, this performs at 425ms for inference per image
- Faster_rcnn_resnet101_ava_v2.1: This model is trained on Atomic Visual Actions (AVA) data source. “The AVA dataset densely annotates 80 atomic visual actions in 430 15-minute video clips, where actions are localized in space and time, resulting in 1.58M action labels with multiple labels per person occurring frequently.” This model is trained using video clips to recognize actions and uses 80 classes of verbs defining actions. Even though it was trained on videos, it can be used for label generation for still images.
- Faster_rcnn_inception_resnet_v2_atrous_lowproposals_coco: This model is trained on MSCOCO data using a set of 90 classes (labels). On average, this performs at 241ms for inference per image
- Ssd_mobilenet_v1_coco: This model is trained on MSCOCO data using a set of 90 classes (labels). On average, this performs at 30ms for inference per image

What did not work

The existing models that are available were developed with a focus on born-digital photographs of everyday objects and scenarios. These models contained objects that were not present in historical photographs. One such example is from an image found at https://collections.lib.utah.edu/ark:/87278/s6z06f6j . That image was submitted to the im2txt script along with a MSCOCO base model, it identified a laptop in the photograph, which is wrong (Laptops were only introduced in 1980s and models must incorporate that knowledge when they output possible labels via object detection). The original model was trained on a dataset that had laptops in the images, which have a similar shape and orientation as a paper. This leads to the determination that the photograph below contains a laptop.

The descriptions of photographs that are currently in the digital collection is heavy on proper nouns. Since most of the images are from local-sources and tied with the history of Utah, the objects were described in detail. For example, the photograph in Fig 2 would be described as **Hyrum Smith Mecham and Jane Harriet Haws Mecham**, rather than two people. A group of people would be described by its context, using the name of an association or an organization that they are part of. When existing models were trained on these collections, the accuracy level of the output decreased due to inconsistent information that was fed to the machine. This drives home an important point – while granular and specific information about a photograph is much appreciated by the user, that data may not be useful for developing models.

Some modifications to the planned process was needed to overcome the proper-noun issue. In this instance, an NLP (Natural Language Processing) script was developed to process existing descriptions and develop a simpler version of the said description. The script was developed using the spaCy framework, and it analyzes and breaks down a sentence into its semantic components. Then it replaces the proper-nouns with appropriate nouns, and puts the sentence back together.

(Fig 2: https://collections.lib.utah.edu/ark:/87278/s6jt34rd)

What worked

We were able to train some models that generated quality metadata to complement the existing information. The example below is a photograph (Fig 3) from the digital collection, and it is described as Campbell's Ferry. The photograph does not have a ferry in it, and the name simply refers to the location where the photo was taken. This image will not come up in the search results unless, the user already knows the name of the location and searching for images with that as a search term. If a user simply wants historical photographs of suspension bridges, and uses it as a search term, they will not find this particular image.

(Fig 3: https://collections.lib.utah.edu/ark:/87278/s6319vk5)

As shown below the machine suggested the following terms for the above photograph. This information could be included as additional keywords for the image, which would increase the findability of that particular image.

- suspension bridge
- pier
- worm fence
- snake fence
- snake-rail fence
- Virginia fence
- viaduct

- steel arch bridge

III. Accomplishments

As we conclude this phase of our project, we are happy to report that we were able to meet and in fact surpass all the project goals that were mentioned in the grant proposal. We developed NLP workflows for data cleanup (released as open source & made available via github). We also developed a prototype for processing the image. The deliverables include three major components.

- Pre-trained models described above
- A Vagrant box including the environment and scripts needed for following tasks
  - Data preparation
  - Training
  - Evaluation
  - Inference
- Prototype of a web application for content modification workflow (screenshot included below, Fig 4)

Fig 4

4. Lessons learned

The following are the lessons that we learned as part of this project: Able to figure out or answer with clarify the question of whether existing ML models work in archival context.

- Existing machine learning models were developed for photographs of the current modern world with everyday objects and scenarios in mind. These models did not translate well for historical photographs.
- The current metadata was using proper-nouns to describe objects. While having granular, in-depth information about an image is always useful from a user's perspective, these terms were not generic enough for the machine to formulate an understanding of the objects. Digital Library data needs to be transformed before they can become useful for new machine learning models.
- The currently available im2txt script[2] was developed for a single GPU for the training. We modified the script to use multiple GPUs. However, given how each step is dependent on the previous step, the script was not utilizing multiple GPUs in its optimal capacity. We were able to achieve better efficiency by training separate models concurrently by allocating each model to its own GPU.

5. What's next?

We are excited to report that we have been able to reach out to multiple organizations to further develop this project. After making substantial progress, to align with the goals, we renamed the project as – "Sheeko: A computational helper to describe digital images". Sheeko means "Story/narration" in Somali, and our goal is to develop models that can generate captions/keywords for images. We are collaborating with Computer Science department at University of Utah to bring in additional machine learning expertise for develop Sheeko. We recently submitted an NEH grant under Advancing Digital Humanities section to further develop this project, and as part of it, created an advisory committee comprising of leaders in this area across the nation.

We are very thankful to the Lyraris organization for giving us the seed funding to initiate this project, and supporting us throughout the development of this project in the initial phase.

6. More information

For additional information on the project, and to follow updates, please visit https://sheeko.org

[2] https://github.com/tensorflow/models/tree/master/research/im2txt