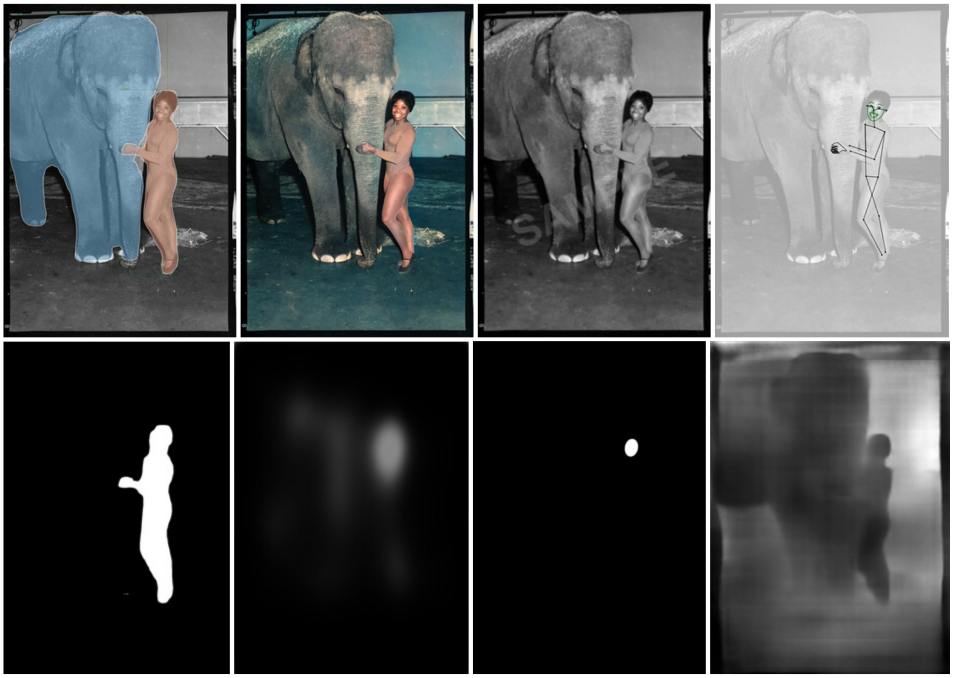
Supporting Cultural Heritage Research in Historic Photography Archives with Machine Learning and Computer Vision



Supporting Cultural Heritage Research in Historic Photography Archives with Machine Learning and Computer Vision

Golan Levin, Professor of Computational Arts, Carnegie Mellon University
David Newbury, Enterprise Software and Data Architect, J. Paul Getty Museum
May 31, 2020

Introduction

In this project, we address the challenges faced in the research and annotation of large digitized photography archives through the development of prototype software tools that employ computer vision, machine learning (ML), and the International Image Interoperability Framework (IIIF). With support from the National Endowment for the Humanities, we studied the Teenie Harris Archive from the Carnegie Museum of Art (CMoA), an extraordinary collection of over 60,000 images of 20th century African American life in Pittsburgh, as the context to help us understand these challenges, and in order to allow us to test our tools on a complex, rich source of images and metadata.

Over the past four years, we have partnered with experts in machine learning and library sciences to develop, test, and document these challenges. We have generated a significant body of data, metadata, code, workflows, and documentation, and have used the results of this analysis to directly augment the Teenie Harris Archive, working with the Carnegie Museum's archivists to both inform their practice and to provide new ways of understanding their collection.

We have discovered that some of our initial assumptions, about where the value of this research would lie, were flawed. In particular, the high rate of change in ML research meant that any specific code that we used and developed, while productive, was destined to be almost immediately obsolesced and replaced by more powerful, easier-to-use tools. We also discovered that the affordances of IIIF, while valuable to this work, are not a perfect match for ML workflows. However, we also discovered through our process and our collaborations with experts that there are a clear set of new algorithmic practices with terrific value for image archives. Although the specific implementations of these tools are certain to continue evolving rapidly, their core affordances represent an important new vocabulary for analysis which, we believe, is essential for archivists to understand.

In this paper, we will describe what we learned through this process, focusing on three core areas:

- What sorts of workflows are possible using machine learning and computer vision with a large historic photography archive, and what kinds of data can result from that effort
- How IIIF can be used, and where it cannot, as part of an archival analysis project using ML

- How to translate the results of a ML workflow back into the daily practice of cultural heritage metadata

In addition, we discuss an interactive visualization system which we developed for the museum's archival team, and which we adapted into a public display at their invitation.

The Teenie Harris Archive

Contents: Images and Metadata

Charles "Teenie" Harris was a photographer for The Pittsburgh Courier, one of the most influential black newspapers of the 20th century. In a career that spanned more than four decades, Harris captured the events and everyday experience of African American life. In 2001, the CMoA accessioned his more than 80,000 photographs and negatives, creating the Teenie Harris Archive. Much of the rich history of Pittsburgh's African American community, from the 1930's through the 1970's, through the Jim Crow and Civil Rights eras, is recorded in this unique and extensive photographic collection.

With previous support from the NEH, archivists at CMoA have digitized the Teenie Harris Archive, and have spent the past two decades captioning, dating, tagging, and adding other metadata to the photographs. (It is important to note that their work is performed in close coordination with representatives from Pittsburgh's African American community, with guidance and oversight by an advisory committee composed of Harris family members, academic specialists, and community leaders who have insisted on the African American community's ownership of the history represented in Harris' images.) Much of the Museum's annotation work has been conducted through interviews collected directly from Teenie Harris' contemporaries, and, wherever possible, with the original community members documented in his photographs.

Challenges in Analyzing the Teenie Harris Archive

In March 2016, the Innovation Studio laboratory at the Carnegie Museum of Art provided our team with a hard drive containing 59,278 high-resolution grayscale TIFFs from the Teenie Harris Archive, organized into 718 directories. They also provided access to an online database with accession numbers, brief textual descriptions, and estimated dates (or date ranges) for nearly all of the photographs. Our understanding was that, at the time that we received the data, some 20,000 additional objects in the Teenie Harris collection remained to be digitized and annotated, possibly including duplicates of the materials we received.

Several challenges specific to the Teenie Harris Archive have shaped the development of our project and its outcomes.

- **Sensitivity of the subject matter.** The Teenie Harris Archive documents real people: numerous still-living people, and many recently-deceased people survived by living

family members. The archive focuses on the lives of black people, and the history of the close-knit black community in Pittsburgh. We acknowledge the fraught histories of the depiction of persons and communities of color in the United States, especially in relation to computing—for example, how machine learning technologies such as face recognition have been used and abused in the contexts of domestic surveillance, carceral geography, and law enforcement. In light of this, many different considerations must be taken with how the Teenie Harris data is labeled, shared, and de-anonymized—with respect to the privacy of depicted individuals, the lens through which a community wishes to be represented, and with respect to the possible perpetuation of racist stereotypes or algorithmic bias.

- **Contemporary biases in ML systems.** It has been said that “the past is a different country”. From the perspective of most machine learning models and categorization tools, which have been exclusively trained on images of modern subjects, the Teenie Harris Archive depicts an anachronistic and sometimes unintelligible world. Cars are differently shaped; people wear unrecognizable hats and furs; homes are outfitted with unfamiliar appliances like wood-burning stoves. Meanwhile, a significant portion of the 80 categories of objects recognized by Facebook’s Detectron algorithm include concept tags like “pizza”, “backpack”, “skateboard”, “laptop”, “cell phone” and “microwave”. While these terms might be useful for describing scenes in a modern college dorm, they yield only false positives when applied to images from the 1940s.
- **Monochromatic images.** The Teenie Harris Archive consists exclusively of black-and-white photographs. However, most machine learning tools for image analysis and recognition have been trained on, and designed to process, modern color images. For algorithms in which the color of an object contributes to its successful detection, the application of such a tool to black-and-white imagery can produce less than satisfactory results. As we discuss later, we produced a novel method for coping with this problem.
- **Unstructured textual metadata.** The bulk of textual descriptions made by the CMOA are unstructured. From a practical standpoint, working with these descriptions has necessitated additional effort, and imposed limitations on what is knowable from them.
- **Restrictions on reproduction.** We (representing Carnegie Mellon University and The J. Paul Getty Museum) don’t have the rights to distribute images from the CMOA collection. Despite working in partnership, we are restricted from making our IIIF services available, and putting high-resolution versions of Harris images online. This has imposed delays and limitations on our ability to share our work with other researchers.

Reproducible Workflows and the Teenie Harris Archive Analysis

One of our initial goals in this project was to create reproducible workflows for historic image archives that leveraged state-of-the-art machine learning algorithms. This proved to be more challenging than we anticipated—not because it was difficult to create productive workflows, but because the hyper-rapid obsolescence of cutting edge tools made efforts toward their exact *reproducibility* almost futile. We created an extraordinary set of image analyses annotations, and

have published the code we used to generate this data on the STUDIO's GitHub account[1]. However, we found that machine learning tools for image analysis are currently evolving so rapidly that the processes we devised became obsolete almost as soon as we finished using them. Better algorithms, but even moreso, significantly easier-to-use processes and tools became available almost immediately.

The underlying tools that we used: Python and Jupyter Notebooks for ML scripting, IIIF for image access, and Processing, OpenFrameworks, and P5.js for sketching and interactive visualization, remain excellent tools, but the specific ML frameworks, commercial application programming interfaces (APIs), and algorithms we used changed dramatically during the course of the project. In a scenario which almost comically repeated itself, we would identify a useful-looking analysis algorithm that had been recently published by some researcher or laboratory; endure the challenges of configuring a Linux computer with the right combination of libraries to get it working; and process the Teenie Harris Archive with the algorithm. Two months later, an update to a component library would break the entire workflow; and meanwhile, an improved and much more user-friendly version of the same algorithm would be released, suitable for use by a layperson in an easy-to-use framework for MacOS, Windows, or a web browser. (In the documentation of our data, we provide links to the improved tools and equivalent workflows that supersede the ones we used.)

What is likely to persist, and what we feel will be most valuable for others working in the field, were the insights that came out around what it means to work with the results of this sort of workflow. The data that comes out of these tools, while wildly varying in quality and semantic structure, falls into a set of patterns that do not align well with traditional archival practice or cataloging. In short, we have identified a set of analysis routines which we anticipate will become part of a standard toolset for large archives (though the specific implementations will change), and which, we predict, will stretch the practices of archival stewardship.

The practice of processing archives as units is familiar to archivists: the work of summarizing and describing collections is core to that practice. However, the summarizations of those materials that come out of ML practices are unfamiliar: they are dense, item-level, and often best represented as graph structures with weighted nodes or as per-pixel annotations that describe statistical properties. The next section provides an overview of these comparatively novel requirements.

Types of Data Structures

As we performed our analysis and reviewed the resulting metadata that was produced by our workflows, seven common flavors of result structures emerged. We have classified these as descriptions, classifications, identified entities, geometric entities, per-pixel maps, networks, and arrangements.

---

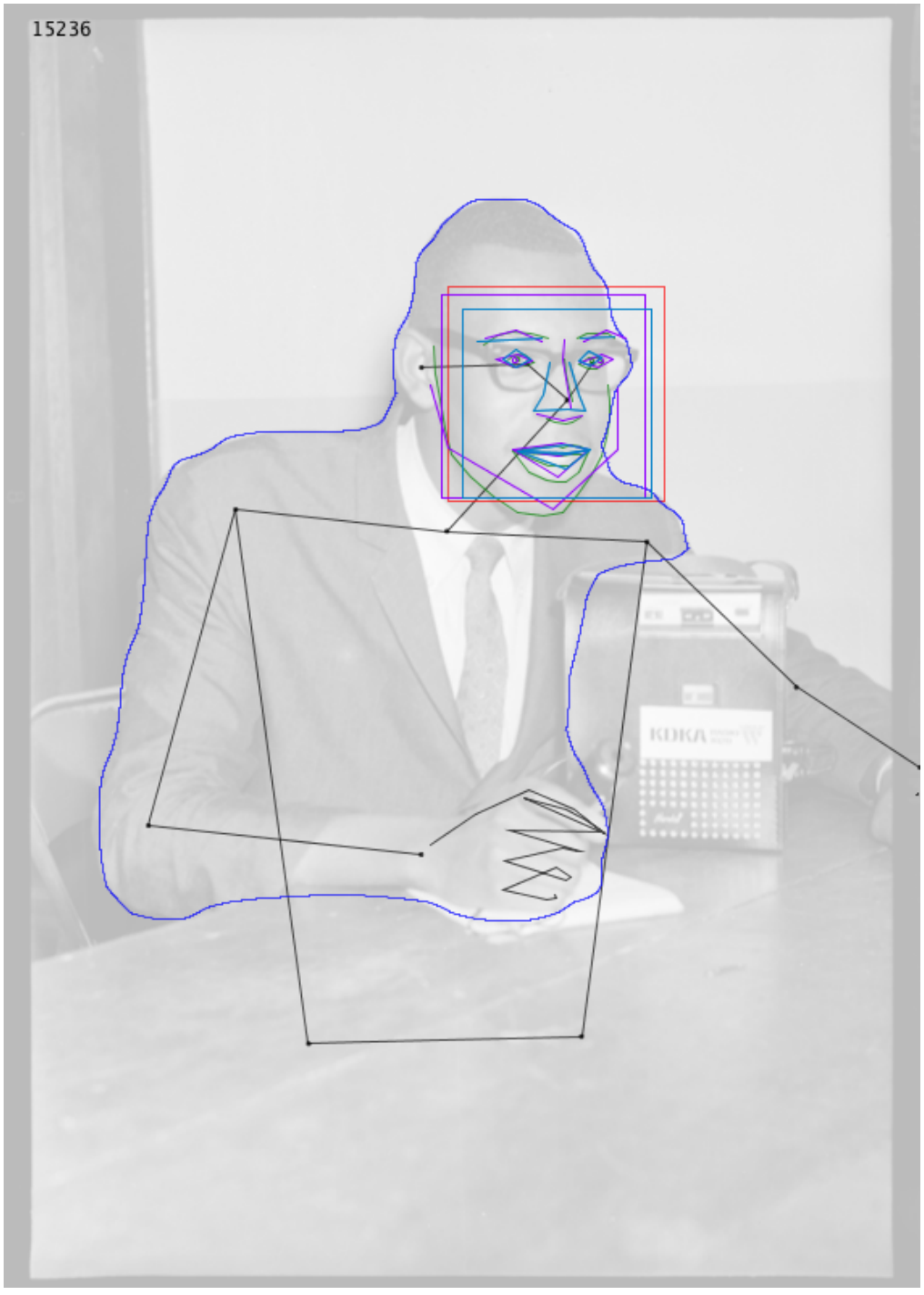
[1] https://github.com/CreativeInquiry/TeenieHarrisProject

1. **Description** is the easiest type of information to handle, and most closely aligned to traditional cataloging practice. Quantitative numbers such as an image's overall luminance, or automatically-generated textual tags, fit well into existing practice.

2. **Classifications** are also well-aligned with traditional cataloging practice. For example, the Google Cloud Vision API provided a list of subject terms that, while not described as Library of Congress Subject Headings, would not be unfamiliar to an archivist trying to understand how to process this data.

3. **Identified Entities** are also not unfamiliar—they map well into the standards of authority files. What is new is the sheer number of entities, as well as the lack of human-understandable description of what those entities represent. The dLib library was able to identify more than 245,000 faces across the dataset, and the OpenFace library also provided methods for calculating correspondences between those faces, but "Face #5000" is not a traditional entry within an authority file, nor is an entry that could be understood meaningfully by a person. This does not mean such entities are not *real,* nor that they're un-useful. However, understanding how to contextualize this information within a representation of archival information is difficult.

4. **Geometric Entities** represent particular regions, locations, or collections of locations within an image. For our project, these included bounding boxes, complex polygons, and skeletons, all of which can be described as sets of 2D vertices lying within the 2D representation of an image. Bounding boxes are simple, axis-aligned rectangles, typically represented by an (X, Y, Width, Height) or (Left, Top, Right, Bottom) data structure. Complex polygons consist of sets of closed shapes; they are conceptually similar but have more complex and variable representation requirements that make both data storage and processing more complicated. (For example, a complex polygon representing a single object could actually be composed of several contours, representing a combination of multiple holes and disjoint parts.) Skeletons consist of sets of 2D vertices representing the pixel locations of canonic body joints and landmarks, such as knees and shoulders, or (on the face) noses and eyebrows, and may have missing or occluded data. Bounding boxes, complex polygons, and skeletons are often described using pixel dimensions that relate to a specific image derivative—but they can be recalculated for any derivative of that image through a process of translation and scaling. While not difficult for a human to understand (we've been drawing boxes around important things forever), they become specific entities themselves within our results—and even their imprecision makes them difficult to represent. For example, we used four different face recognition algorithms (Google Cloud Vision API, Microsoft Cognitive Services, OpenPose and OpenFace), and these systems were in broad agreement in detecting certain regions as faces. However, for any given face, the *precise* localization of the bounding boxes provided by the four different algorithms had meaningful variation. It requires special handling to define the correspondences between those faces, and to tie those faces back to the entity being identified.

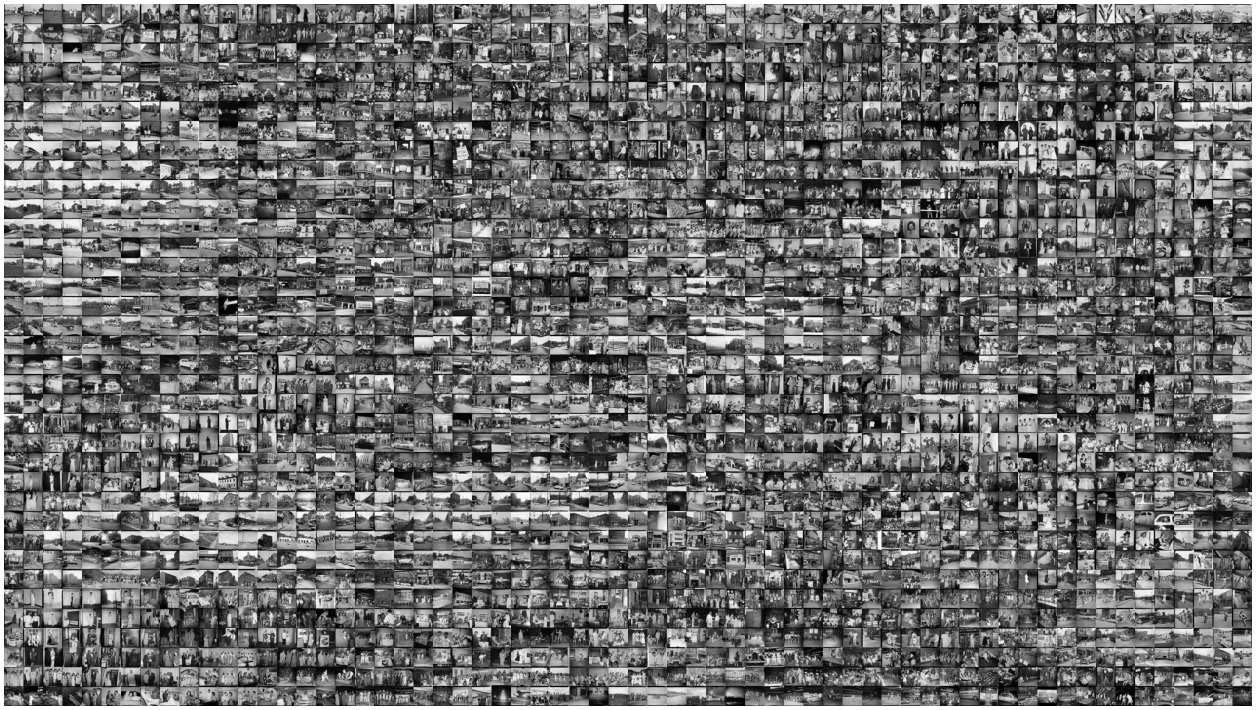


The screenshot above depicts some of the geometric entities that we calculated for image #15236 from the Teenie Harris Archive. These include body contours from Detectron (blue), body skeletons from OpenPose (black), face details from OpenPose (green), face bounding boxes from OpenFace (red), face landmarks and contours from Google Vision API (magenta), and face data from Microsoft Cognitive services (cyan).

5. **Per-pixel maps** are arrays of data—technically, “images”, such as heat-maps—where each pixel in the input image has a corresponding quantitative datum in the output “image”. Per-pixel maps are not common in cultural heritage metadata: they do not correspond well to how archivists traditionally represent knowledge about images. Sometimes, a per-pixel map (such as a depth-map) can be perceptually legible to a viewer as a version of the original image. In other cases, a per-pixel map (like a heat-map) may represent totally abstract data about the original image, and may not be legible on its own. In our work with the Teenie Harris Archive, we used algorithms that produced per-pixel weightings including “Saliency” (a heat-map representing the likelihood that a given pixel might be perceptually interesting, based on an AI eye-tracking model), “Estimated Depth” (an image representing an estimate of how far each pixel is from the camera), and “Speculative Chroma” (an AI’s guesses about what the original colors in a black-and-white image might have been: grass is probably green, etc.) Conceptually and technically, per-pixel maps are best treated as new “channels” of information, and there are some valuable parallels to multispectral imaging here. However, per-pixel maps—especially when they contain statistical estimates of likelihood derived from machine learning algorithms—may represent concepts that are much more abstract than “red”, “x-ray transmissivity”, or “presence of cobalt pigment”.

6. **Networks** are graphs that represent directed relationships between single entities, typically with a weight or other annotation assigned to the relationship. These entities might be the images themselves, or might be other entities depicted within the images, such as faces or people. In archives containing tens of thousands of documents, the density of results that can result from these analyses is potentially very large. In the approximately 60,000 images of the Teenie Harris Archive that we analyzed, for example, we discovered more than 245,000 faces—resulting in billions of possible relationships. In our work, two classes of graphs are represented: sparse graphs, where each connection is likely of high value (for example, the results of facial recognition, where each edge represents a likely match between two images of the same person) and dense graphs, such as similarity matrices, which result from computations of image similarity. Individual relationship weights are often not specifically meaningful, except in comparison to others.

7. **Arrangements** (also called *embeddings*) are collection-wide spatial mappings in which every photograph in the collection is assigned to a different cell in a 2D grid. Typically considered a secondary data product when calculated from dense relationship networks, arrangements are nevertheless valuable data—able to reveal internal structures within the collection itself, such as clusters of similar photographs, that may not otherwise be obvious.



The screenshot above shows an example of an "arrangement". Here, 2304 images from the Teenie Harris Archive (approximately 3.9% of the total collection) have been arranged into a 2D grid according to their visual similarity. Within this arrangement, it is possible to discern clusters or neighborhoods of similar images, such as the group of street scenes at upper left.

Identifiers for the Identification, Reconciliation, and Concordance of People

Each of the seven types of data described above represent new knowledge about the archive that must be stored in order for a workflow to provide utility down the road. This knowledge must also be related back to (A) the digital images themselves, then back to (B) physical objects such as negatives or prints, and then finally to (C) the real world entities that are depicted in these representations. Understanding which of these three subjects each generated dataset describes is essential, and the richness of the metadata that emerges from these practices often demonstrates how our current practices for metadata analysis and storage elide the differences between these representations and rely on our human assumptions to properly relate the data to the appropriate level of abstraction. That reliance works at the scale of human-generated metadata, but falls apart when dealing with the incredible volume of information that ML processes are capable of generating.

For our work, none of the analysis that we generated relates directly to the physical material (though there are examples of work in this field that do[2]). Instead, our results applied to either the images themselves, or the entities depicted within them. To manage this, we needed to

[2] One good example of physical analysis would be D. H. Johnson, E. Hendriks, M. Geldof, and C. R. Johnson, Jr., "Do Weave Matches Imply Canvas Roll Matches?," *38th Annual Meeting of American Institute for Conservation of Historic and Artistic Works*, Milwaukee, WI, May 2010

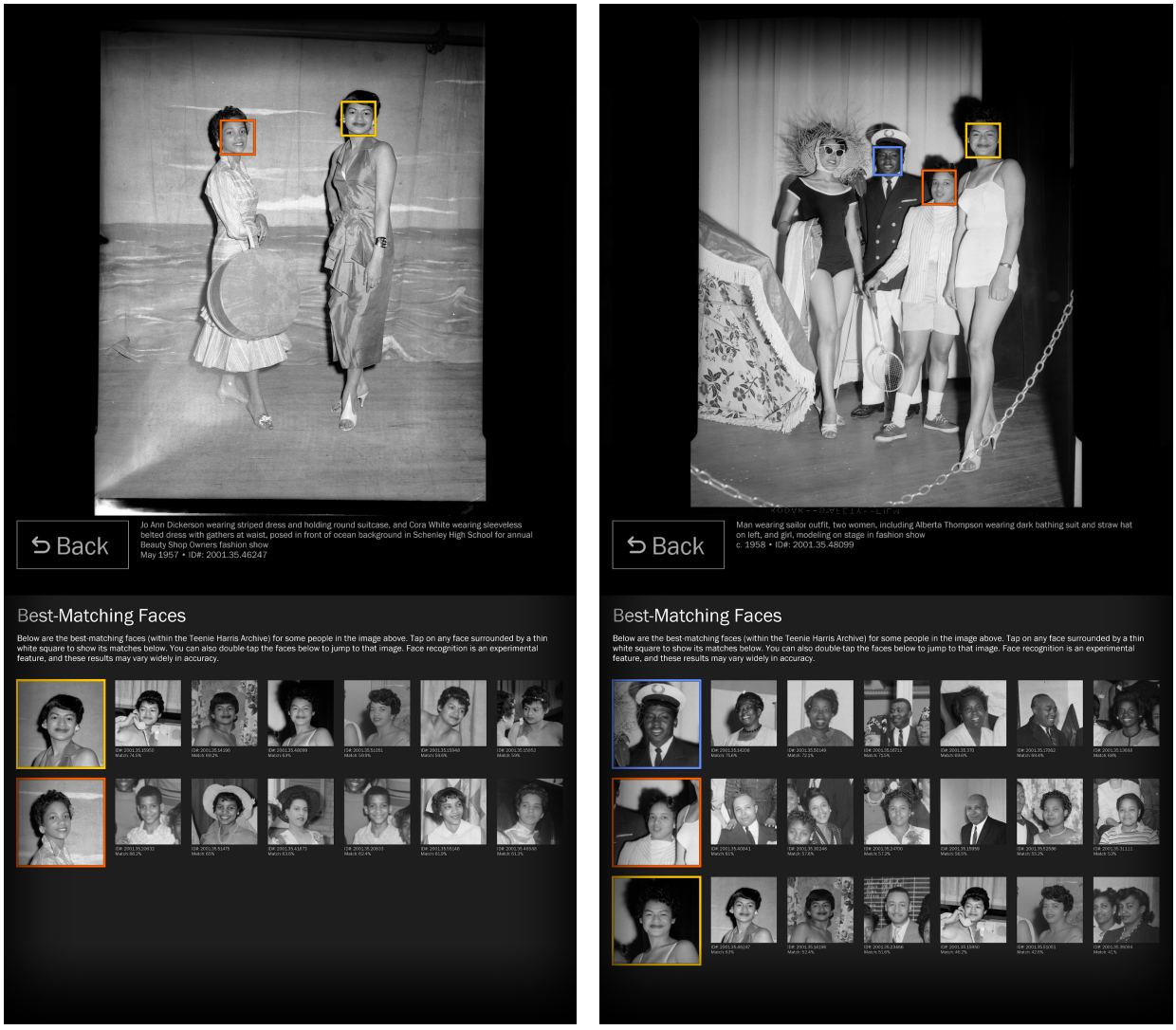
create additional identifiers for these conceptual entities. We generated a canonical identifier[3] for each image: an identifier for the image itself, different from the filename on disk. We also learned that many of the analysis tools that we used were not capable of carrying this identifier forward into their results—instead simply producing a list of results, one row per image. Using this canonical identifier not just to identify, but also to provide an explicit order for the images allowed us to treat an ordered set of results as having an implicit relationship to our identifiers, essential in allowing these results to be connected back to the identifiers and thus the images.

We also generated identifiers for faces within each image, connecting faces from result sets from different ML toolsets and workflows. This also allowed us to connect a “computed face” (that is, a representation of a face as a region of pixels within an image) to a “conceptual face” within a single image—that is, the aggregated knowledge about a face, reconciled across several different analyses of that image, to an inferred “real world face”, the representation of a face of a person in the world. Each of these faces exist within our result set, and each is subtly different, highlighting both the complexity of the information within the result set and the difficulty of communicating the nuances of these representations to others who are interested in using this data down the road.

There is also the problem of reconciliation here, and of communicating to users of this data what the reconciliation is and how it was computed. To reconcile computed faces, we relied on code we wrote to compare the geometric entities that resulted from each workflow with other workflows, and used those to compute a “conceptual face” geometrical entity based on criteria for proximity and overlapping. No single one of the computed faces would be “correct”, but several of them, together, might form a useful conceptual face, for which it is useful to create an identifier.

Once we had produced a dataset of conceptual faces, we could treat those regions of pixels as unique images and use a similarity metric to group these conceptual faces together—assuming that faces that the computer finds to be similar are *possibly* representations of the same real world face, and thus the images *possibly* depict the same person. For the Teenie Harris Collection, this is not a bad assumption: across the 60K images we analyzed, our workflows identified 245,000 conceptual faces. During the decades when the Teenie Harris photographs were taken, the total population of Pittsburgh was approximately 500,000 people, and the African American population of Pittsburgh was much smaller. In short, there are likely not 245,000 unique people depicted in the archive. Although it is difficult to be certain, we estimate it to be more likely that there are only 15,000-30,000 unique people depicted in the Archive, with some individuals (local politicians and socialites, acquaintances of the photographer, etc.) appearing hundreds of times.

[3] Canonical_filename_order.txt within our Github repository contains this list of cross-references between the original file name as provided from CMOA and our assigned identifier.



The screenshots above illustrate how our workflow can facilitate a human archivist in the task of identity chaining. The Museum's caption for the image at left describes the scene: "Jo Ann Dickerson wearing striped dress and holding round suitcase, and Cora White wearing sleeveless belted dress with gathers at waist". Our system has detected the faces of the two women in the photograph, and has indicated one of their faces with a yellow square. Although our system is unable to infer which of those faces belongs to Cora White, this can easily be done by a human archivist from a close read of the caption. Simultaneously, our system has automatically recalled a series of six other photographs that appear to include a face that matches the one in the yellow square. Selecting the third option, we arrive at the image at right, of a "man wearing a sailor outfit, two women, [...] and girl, modeling on stage in fashion show." An archivist can now decide, based on a close examination of faces: is the same person depicted in both photographs? If so, then Cora White is the woman on the right in both images. Our system provides tools for making and recording such determinations.

Our workflow does not do the work of definitively determining that two faces are the same person. Instead, it provides us with a statistical likelihood that any two faces within our collection are the same person, resulting in billions of possible matches, many of which are extremely improbable. One of the skills and responsibilities of the human users of a system like this is to

determine what we will use to perform that sort of reconciliation. There's no good correspondence to "geometric overlap" that applies to this sort of analysis, but limiting the number of relationships is an essential step in generating a useful dataset. In developing an interactive interface for the museum's archivists, we decided to present only the top six most-likely matches to other faces. (Additional prospective matches can be pulled from the database if requested.) This was not an objective calculation; instead it was a subjective judgement based on our skill and experience using these types of collections. Now let us suppose an archivist decides that two faces represent the same person. There is not a clear way to document this decision within the data: instead, we rely on the data dictionary we generated alongside the data to describe this decision to help others understand that this decision was made. This presents a problem, however, if this data is to be re-ingested back into source systems at CMOA. It is already challenging to capture annotations which are not definitive, but have (say) 40% accuracy; how can the Museum additionally capture the pair-wise probability that two faces match, separate from the accuracy of their image-to-entity relationships? Just as it is not practical for the Museum to store billions of potential matches, their systems also do not allow the imprecision of this information to be captured as well.

There are additional concerns that arise with this work—our workflows do not have the ability to exercise *judgement* about the connections that are made. In a manual review of the data, we discovered an image of a woman who is named in the associated metadata provided by CMOA. Our facial similarity analysis discovered a highly-similar match for her face in another image within the dataset—a photo that depicts an anonymous nude woman. A human cataloguer might have decided for reasons of privacy to not explicitly connect these two images together, even if they'd noticed the connection, but the computer does not have the ability to do so. The number of connections, even if we are only looking at the top six, is still in the millions—it is not practical for each connection to undergo human review. And the ethics of this de-anonymizing is subjective: a family member may not want their name connected to images depicting someone committing a crime, but a family member of a victim might very much want that connection to be made. There is not a practical way to encode this sort of logic within the reconciliation logic. The work of deciding what rules will be used to describe links requires sophistication both in understanding what the tools may produce, and what the human implications of making a connection might be.

Summary of Our Analyses of the Teenie Harris Archive

The first phase of our project, Image Dataset Analysis, involved the application of a battery of standard open-source and commercial computer vision algorithms to the Teenie Harris image dataset. To begin with, we analyzed the Harris archive with several robust commercial APIs including Google Cloud Vision, Microsoft Cognitive Services, and Imagga. Using these commercial tools involved sequentially uploading Teenie Harris images to an online, cloud-based API endpoint, and then downloading the resultant analysis files in JSON format. These cloud-based services were either free to use, or low in cost; for example, analyzing 59,278 images with the Imagga library cost about $80.

Following this, we processed the Teenie Harris Archive with “research-grade”, open-source code libraries including OpenPose, OpenFace, Detectron, FCRN Depth Prediction, SalGAN, and the Inceptionv3 and VGG16 discrete convolutional neural networks (DCNNs). The code for these tools was downloaded directly from the GitHub repositories of individual labs and researchers, and then compiled and executed locally on a Linux desktop with powerful GPUs.

In addition to the above, we fed the results of these analyses into the UMAP dimensionality reduction algorithm to calculate a wide range of secondary 2D *embeddings*, or spatial arrangements, of the images in the archive. These arrangements organize the entities in the archive into spatial clusters according to their similarity, allowing us to derive and visualize “maps” of the archive’s contents.

Finally, the datasets representing the analysands and embeddings from these processes have been organized and documented, bundled into downloadable archives, and published to inexpensive long-term online storage and public GitHub repositories.

Metadata and other Data Products

In this section, we list the metadata, derivatives, and other data products that we produced during the course of this project, in order to analyze the photographs in the Teenie Harris Archive.

- **Google Cloud Vision API:** We used the commercial Google Cloud Vision API[4] to provide information about the Teenie Harris Images including facial landmarks, facial expression analysis, object recognition, and optical character recognition (OCR) of any text in the scene.

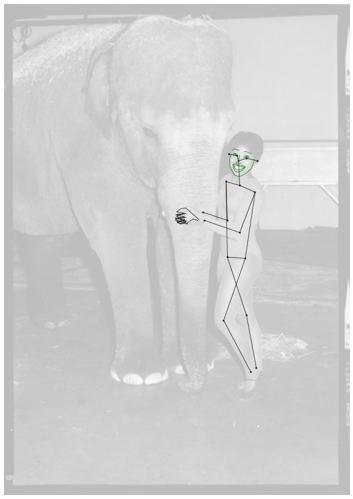
- **Microsoft Cognitive Services API:** We analyzed the Teenie Harris Archive with the commercial Microsoft Cognitive Services API, which provides information including face landmarks, gender estimation, age estimation, and facial expression ("emotion") estimation[5].

- **Imagga API:** the commercial Imagga image analysis service provided a wide range of semantic descriptors and confidence values, primarily for object detection[6]. For the most part, these proved to be redundant with those provided by Google and Microsoft. The Imagga service also reported editorial and thematic evaluations (e.g. “sexy”, “macho”, with accompanying confidence factors) which we found less than helpful for our application.

[4] https://cloud.google.com/vision/
[5] https://azure.microsoft.com/en-us/services/cognitive-services/
[6] https://imagga.com/

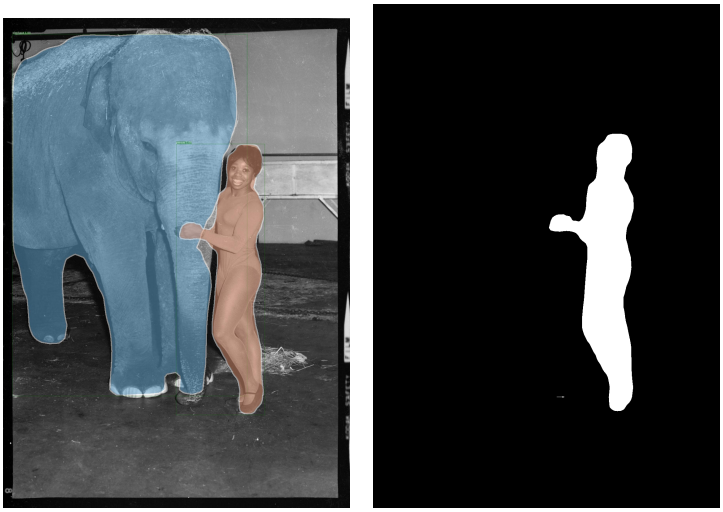
- **Luminance Metadata:** In our conversations with the CMOA archivists, it emerged that one of the ways they recall an image is whether it depicts an overall dark or light scene. To support this, we produced a JSON file containing statistical information about the brightness of every image in the archive, including, as individual numbers: the average grayscale value; the median grayscale level; and the standard deviation of grayscale levels in the entire image. This information is relatively easy to calculate.

- **Skeletons (via OpenPose):** OpenPose is an open-source library that estimates the stick-figure-like "skeletons" of images of people[7]. The data produced by this library includes 2D body skeletons, facial landmarks, and the joint locations for hands. We used the CMU Perceptual Computing Lab's OpenPose version 1.2 in late 2018—but much easier-to-use implementations of this algorithm can now (2020) be found in tools like Runway.ml and ml5.js. We used OpenPose to calculate the skeletons across the entire Teenie Harris Archive, producing a set of 59,278 JSON files.



- **Human and Object Contours (via Detectron):** "Detectron" is an open-source library for object detection produced by Facebook[8]. Detectron not only reports the (likely) presence of up to 80 different categories of objects (such as cars, certain kinds of animals, and people); it also provides 2D contours that indicate which pixels belong to a detected object. (We used Detectron v1; which is now deprecated; the much-improved Detectron2 has recently become available[9], and easier-to-use versions of this library can now be found in the RunwayML service.[10]) We produced 58,696 JSON files from the Teenie Harris Archive, each containing contours of objects (and their category labels). In addition, we also generated binary (black-and-white) pixel maps indicating the bodies of detected people.

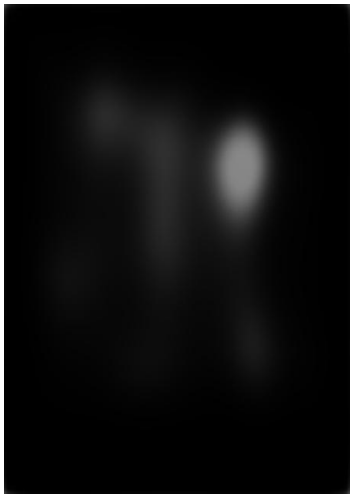
[7] https://github.com/CMU-Perceptual-Computing-Lab/openpose
[8] https://github.com/facebookresearch/Detectron
[9] https://github.com/facebookresearch/detectron2
[10] https://runwayml.com/



- **Depth Prediction (via FCRN):** Recent advances in machine-learning-based monocular depth prediction make it possible to estimate the foreground and background of an image. We used the FCRN Depth Prediction algorithm[11] ("Deeper Depth Prediction with Fully Convolutional Residual Networks", 2016) to produce an archive of 59,278 .PNG images representing the estimated "depth" of the scene in every Teenie Harris photograph. In these images, light colors represent pixels that are estimated to be further away from the camera. A similar and easier-to-use algorithm (DenseDepth by Ibraheem Alhashim) is now provided by RunwayML, and supersedes FCRN. In combination with face or body detection, a depth map can be useful in determining which people are in the foreground of an image, and therefore more likely to be the compositional “subject” of an image.

[11] “Deeper Depth Prediction with Fully Convolutional Residual Networks” by Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, Nassir Navab. *2016 Fourth International Conference on 3D Vision*, 2016.

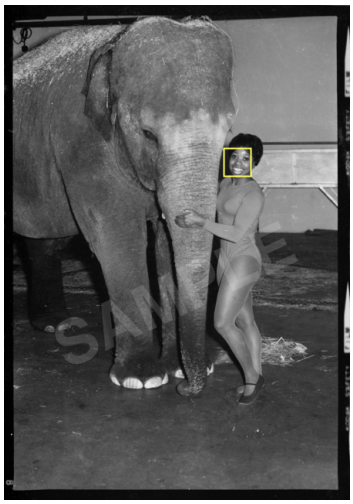
- **Saliency**. We used the SalGAN neural network[12] ("SalGAN: Visual Saliency Prediction with Generative Adversarial Networks", 2017) to estimate the pixelwise "saliency" of every Teenie Harris image. SalGAN has been trained using eye-tracking data, and produces heat-maps that tend to prioritize likely features of visual interest such as faces and text. Saliency information can be useful in making automated crops of image that are superior to center-cropped squares (e.g., for the purposes of a web interface).



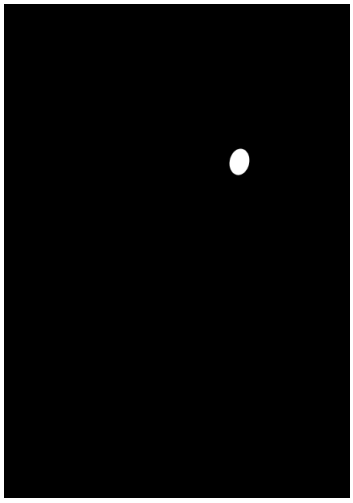
- **Saliency Maxima:** Using the SalGAN results, we computed the local maxima points of the saliency images using the Non-Maximum Suppression algorithm, and stored these as numerical data. Saliency maxima are useful for knowing where points of likely visual interest are in an image, especially in cases where faces and text go undetected. We encoded these locations as sequences of triplets (X,Y,R) representing the center and radius of circles containing (likely) interesting features.

[12] Junting Pan, Cristian Canton, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol and Xavier Giro-i-Nieto. "SalGAN: Visual Saliency Prediction with Generative Adversarial Networks." *arXiv*. 2017. https://github.com/imatge-upc/saliency-salgan-2017

- **Face Rects:** We produced numeric data storing the coordinates of rectangles that indicate the locations of faces in the Teenie Harris photographs. The face rectangles are derived from the union of Google, OpenPose, OpenFace and Microsoft (whichever has data). Where different APIs have produced small differences in detected rectangles, the rectangles stored here are an average of their results.



- **Face Ellipses:** We produced pixel maps containing white oriented ellipses (on a black background) that indicate the presence and locations of faces in each Teenie Harris photograph. The face locations and dimensions are taken from a mixture of Google, OpenPose, OpenFace and Microsoft (whichever has data). The face *orientation angles* are taken from Google API or Microsoft API, whichever has data.



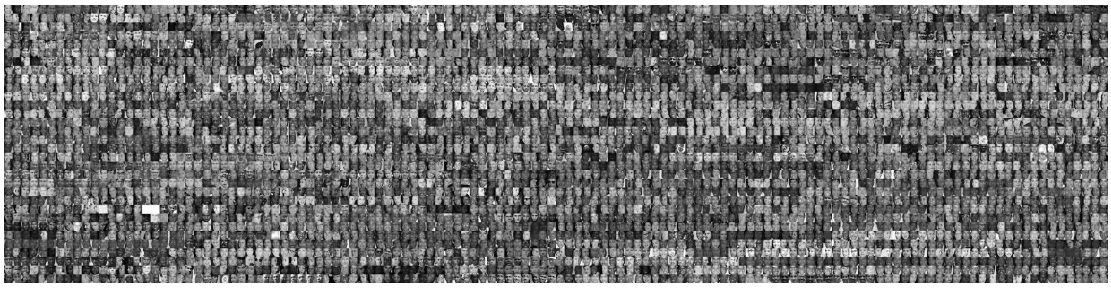
- **OpenFace:** OpenFace is a Python and Torch implementation of face recognition with deep neural networks. Faces are described with a bounding box, and a 128-dimensional vector of face-specific floating-point numbers. These numbers do not indicate geometric information, but rather, locate each detected face in a 128-dimensional “abstract face space” in which similar-looking faces are expected to have proximal positions. These

128-dimensional vectors form the foundation for all of the subsequent face recognition in our project.

- **OpenFace+Microsoft:** We merged the 128-dimensional abstract face descriptors produced by OpenFace, with the gender and age information computed by Microsoft Cognitive Services API. The combination of these data were useful for supervised UMAP/t-SNE analysis of the faces in the archive.

- **People-in-Images:** We produced a set of 59,278 JSON files (one for each Teenie Harris image) containing arrays of people, where each “person” is a concordance of spatially-aligned data from Google, Microsoft, OpenPose, OpenFace, and Detectron. The data includes face landmarks, body contours, and face description data, among other fields. In this data, face bounding boxes from all four services have been checked against each other and aligned, and grouped by person. Note that not every person object contains data from all four services; for example, sometimes a face is detected by one service, but not by another.

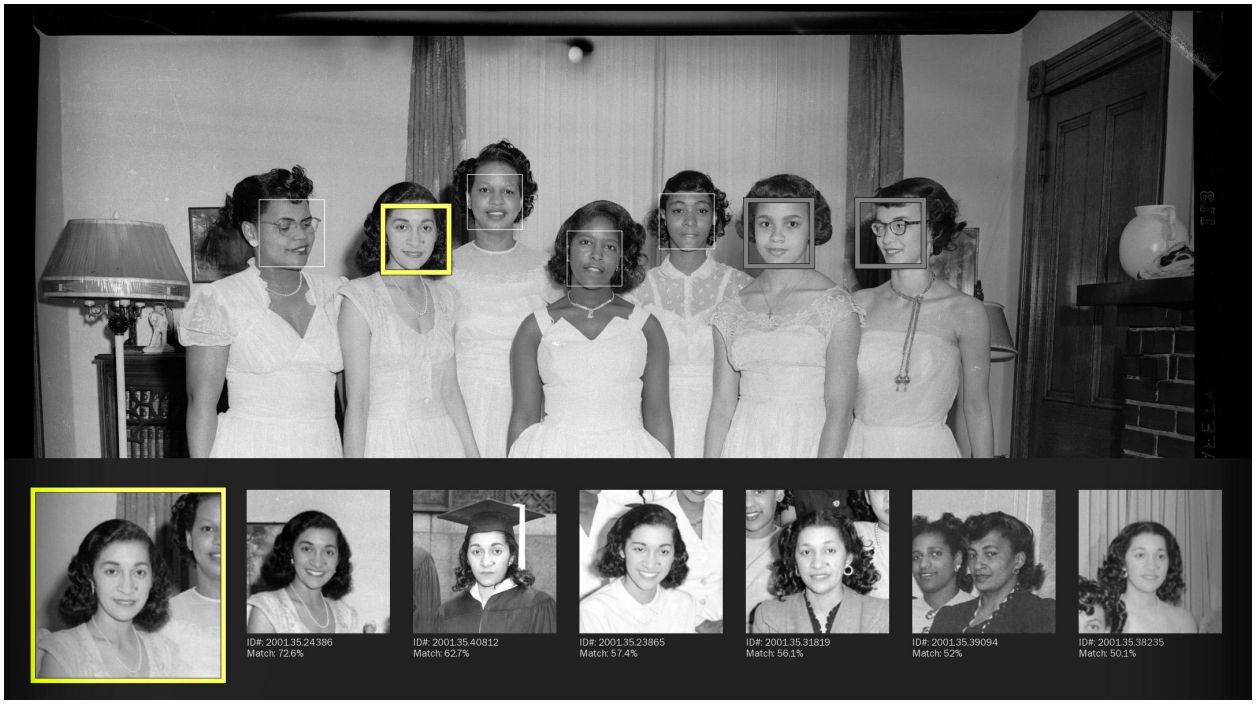
- **Face Metadata:** Derived from the face detection, this JSON file contains aggregate information summarizing the set of faces in each image, the area they occupy, and their (computationally estimated) ages. In particular, this file encodes the number of faces detected in the scene; the average age of all the detected faces; the age of youngest face in the scene; the age of oldest face in the scene; the median age of people in the scene; the age of the largest face in the scene; the area of the largest face in the scene, and the percentage of the image's total area covered by faces. This data represents the union of faces detected by the various services and libraries (Google, Microsoft, OpenFace, OpenPose).



This image shows 4096, or about 1.6% of the more than 245,000 faces we detected in our subset of the Teenie Harris Archive, in accession order.

- **Face Nearest Neighbors.** The set of 59,278 Teenie Harris images we analyzed contains approximately ~245,000 faces. Many individuals are represented several times in the collection. To identify instances in which a person appears in more than one image, we used the 128-dimensional feature vectors produced by OpenFace/dLib, which describe most (though not all) of our detected faces. To identify the closest-matching faces for a given face, we searched for the faces with the smallest Euclidean distance in

128-dimensional “face space”. From this, we produced a set of JSON files which encode a sparse graph about faces whose vectors match up particularly well. Note that there is no guarantee that two faces with closely-matching feature vectors actually represent the same person.



In this screenshot from the visualization tool we developed for the CMOA, the face of a young woman in the upper image is used as a query into the Teenie Harris “Face Nearest Neighbors” dataset, as indicated by the yellow square. In this interface, our system returns the six best-matching faces from other images in the archive. Note that the faces returned by the system include images of the same person photographed under different lighting conditions, with different facial expressions, different clothes (wedding, graduation, etc.), different hairstyles and earrings, different locations, and even at very different ages. Only one of the six faces displayed by the system depicts someone who is clearly not the query individual. The Museum’s archivists can use this interface to approve or reject matches, and help chain identities across the collection.

- **2D Spatial Embeddings/Assignments from DCNNs + UMAP.** One of the most important, prevalent and now-standard patterns for image archive analysis with machine learning is a workflow that combines deep learning and statistics to create a 2D assignment grid for the archive. Used in (for example) PixPlot[13] by the Yale DHLab and the “t-SNE viewer demo” in Gene Kogan’s “Machine Learning for the Arts” (ML4A) curriculum[14], this pattern operates as follows:

[13] https://dhlab.yale.edu/projects/pixplot/
[14] https://ml4a.github.io/guides/ImageTSNELive/

- A pre-trained convolutional neural network (CNN) is used to characterize each of the images in the archive with “perceptual feature vectors”. In our case, we analyzed the Teenie Harris images with the Inceptionv3 CNN[15] trained on the ImageNet dataset. Usually, the CNN process produces a list of 1024 classifications or “predictions” for each image: numbers that describe the likelihood that the image contains a “dog”, “car”, “house”, and many other categories. However, for our feature vector we extracted the CNN’s “second-to-last layer”—a set of 1024 numbers that describe the likelihood that the image contains perceptual sub-components like eyes, wheels and windows. Each of the nearly 60,000 Teenie Harris images was represented with a row of 1024 numbers.

- To these vectors, we then applied the Uniform Manifold Approximation and Projection (UMAP)[16] dimensionality-reduction algorithm, by McInnes et al., in order to reduce these vectors to a representation with just two dimensions. (In earlier implementations of the project, we also used the older t-Distributed Stochastic Neighbor Embedding (t-SNE)[17] algorithm, by Laurens van der Maaten, for this same purpose.) This 2D representation is called an “embedding”, and it makes clusters latent in the data very clear.

- We then processed the 2D embeddings produced by UMAP with custom “Create Grid from Embedding” code by Kyle McDonald[18], in order to produce a 2D “assignment”.

The results of the above process was to produce gridlike mappings of the entire archive. We developed a custom application in C++, using openFrameworks, to speedily visualize the entire archive, as a grid of 214x277 (59,278) small thumbnails images. Results of this are shown on the following page.

In addition to using the Inceptionv3 CNN to produce feature vectors, we also experimented with the use of the Museum’s textual descriptions as a different basis for creating an assignment. For this purpose, we processed the museum’s descriptions through TF-IDF[19] (to identify unique words) and Word2Vec[20] (to generate a high-dimensional description of text) in order to derive quantitative feature vectors from pure text.

[15] https://keras.io/api/applications/inceptionv3/
[16] https://pair-code.github.io/understanding-umap/
[17] https://lvdmaaten.github.io/tsne/
[18] https://github.com/CreativeInquiry/TeenieHarrisProject/tree/master/notebooks
[19] https://en.wikipedia.org/wiki/Tf%E2%80%93idf
[20] https://en.wikipedia.org/wiki/Word2vec

The screenshots above show different neighborhoods within the same (much larger) 2D arrangement: in this case, the Teenie Harris Archive organized spatially according to visual similarity, as determined by th Inceptionv3 convolutional neural network and UMAP. At left is a neighborhood in the arrangement that consists of portraits shot in Teenie Harris's studio, in front of a circular card he used for the purpose. At right is a neighborhood in the same arrangement showing a cluster of images of grisly car crashes, which Harris photographed as part of his local 'beat' for The Pittsburgh Courier.

We verified the utility of our arrangement workflow on an entirely different dataset of images. In October 2019, our team was invited to the National Gallery of Art (NGA) in Washington, DC to participate in its first-ever "datathon"[21]— an effort to use computational tools to analyze, contextualize, and visualize its permanent collection data. We used this opportunity to see whether the Inceptionv3/UMAP workflow we had developed for the Teenie Harris Archive could be easily applied to an entirely different dataset. Instead of analyzing 60,000 black-and-white photographs, we applied our tools to more than 90,000 paintings and prints. The effort was

[21] https://www.nga.gov/press/2019/datathon.html

successful, producing first-ever comprehensive overviews of the NGA collection, such as the image below, an arrangement of all pre-1700 paintings in the National Gallery of Art:

The IIIF Image API and ML Processing

The International Image Interoperability Framework (IIIF) is a set of APIs that provide a standardized method of describing and delivering images over the web. As part of our research, and motivated by ongoing trends in archive stewardship, we investigated the potential utility of IIIF in machine-learning workflows for analyzing the large-scale Teenie Harris Archive. We determined that many of the methods we used to analyze the archive require fast local access to the entire archive at high resolution (e.g. for face detection, body contour detection, and analysis with convolutional neural networks). Other methods we used require simultaneous access to large amounts of abstract data about every image simultaneously (e.g. the UMAP/t-SNE algorithms for dimensionality reduction). Unfortunately, neither of these requirements lend themselves particularly well to IFFF workflows.

For the purposes of this investigation there are two relevant IIIF APIs: the Image API and the Presentation API. The IIIF image API[22] is not a file format or data format for pixel data; it is a set of standard URL patterns a HTTP server can implement that provide HTTP access to specific image derivatives[23]. It also provides a sidecar metadata file that describes technical metadata about those images, allowing software agents to determine which derivatives can be requested without needing to download an image. The benefit of this standard is that interfaces that require image data can develop a consumer using the IIIF Image API and that consumer can then access image files from many image providers without writing custom code for each provider.

This is extremely useful as a tool for building image-access software interfaces that are designed to be reused, or where the tool is built by one party who does not have prior

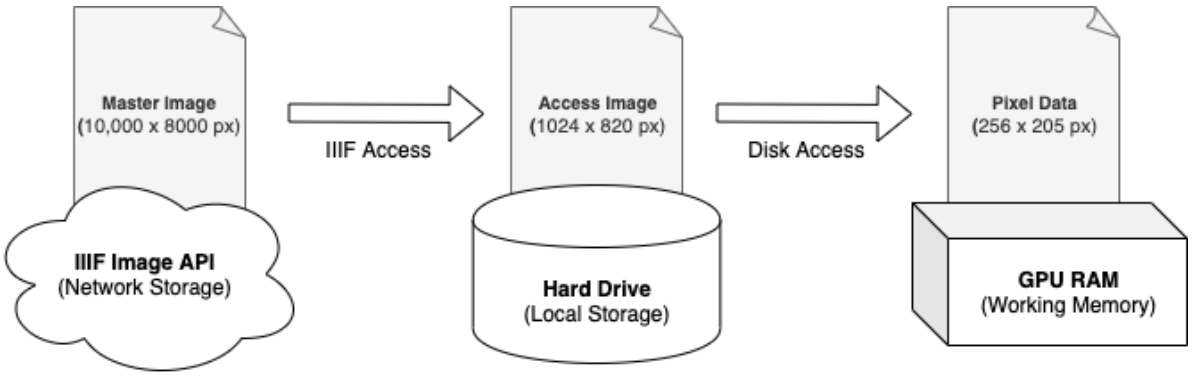
[22] https://iiif.io/api/image/2.1/

[23] An image derivative is a “cropped, scaled, and/or pre-processed version of image”. For example, a 256x256 pixel, center-cropped, black and white JPEG would be a specific derivative of an image. One of the complications that this work brings to light is that we don’t have language that describes the original image--that image is both simultaneously a conceptual work and a specific derivative itself. This is not a unique problem: Functional Requirements for Bibliographic Records (FRBR) is a conceptual model that tries to provide similar language within the bibliographic and library fields.

knowledge of where and how that image will be used. It has a major constraint though: it's designed to provide network access—and network transfer is one of the slowest processes in computing. The speed of network access limits the direct use of the IIIF within machine learning processes. ML typically relies on direct memory access to pixel data, which is often a million times faster than network access[24].

Additionally, it is important to note that essentially every machine learning algorithm already includes (multiple) stages of image resizing and cropping—an operation which is accomplished in a single line of Python code. It is impractical to outsource this image resizing to an IIIF request, not only because of the network delay, but also because many ML processes create and resize versions of images with *many* more than just the usual 3 (R,G,B) channels. Python makes resizing images with dozens of channels trivial; IIIF, not so.

It is better to think of the IIIF Image API as a pre-processing step for generating data sets that can subsequently be subjected to machine learning. In particular, IIIF can be used to standardize the process of creating a set of derivatives on disk that are then loaded into memory and processed using ML. The resizing capabilities provided by IIIF mean that those copies on disk can be obtained over a network connection already reduced to a smaller "access master" size, decreasing the download time and storage needed for those images. Each ML workflow can then process that access master for their specific analysis.



The IIIF Image API also provides a globally unique identifier for both an image and a specific derivative for that image. As we run our workflows a consistent way to refer to those images within data stores is invaluable.

We can also use IIIF's capability for providing on-the-fly derivatives to provide human access for interpreting the results of the ML process. Often the ML analysis will provide data that only makes sense when visualized against the image itself, and the access image used to generate

[24] 150 nanoseconds is a reasonable estimate of the latency in accessing memory, compared to 150 milliseconds for network access. Bandwidth is also a concern, but even the time to first byte difference means that network access is impractical for direct ML analysis. For a good visualization of this difference in speed see https://colin-scott.github.io/personal_website/research/interactive_latency.html

that analysis may not be optimal for human viewing. Being able to easily request a version of the image with a known spatial relationship to the processed derivative at a size and quality optimized for human legibility makes the presentation of these results much simpler, particularly if related metadata also has spatial characteristics. This can also help with rights restrictions--distributing a URL to an image that is hosted on a server owned by the copyright-holding institution is often more acceptable than providing direct access to a copy of the image.

The IIIF Presentation API and ML

There is a second API available within IIIF: the IIIF Presentation API[25]. This API is not designed to hold semantic data about that content or to be used as a metadata exchange format; instead, it describes a method of presenting images, to aid software interfaces in displaying a specific user experience. To be precise, it encodes a machine-readable presentation of 2D content along with contextualizing metadata, including a description, links, grouping and order.

The ability to add a description is obviously useful—titles, rights information, authorship, and other textual information that allows a human to understand what they're seeing is valuable. It is tempting to use this to also capture and present machine-actionable data—but that is explicitly out-of scope for IIIF. Instead, the IIIF specification provides the *seeAlso* property, described in the documentation as: "A machine-readable resource [...] that is related to the current resource that has the *seeAlso* property." This allows the presentation to provide access to semantic metadata.

IIIF's grouping function is potentially enormously helpful. Much of the value of the processes that we generate result in the identification of *sets* of images, and the identification, and more importantly, the presentation of these sets, can be significantly improved through the use of the IIIF Presentation API.

Beyond the contextualizing metadata and structure, IIIF also adds in the concept of a virtual canvas—an abstract spatial region onto which multiple data representations can be, in IIIF terminology, "annotated". In almost every use of IIIF, there is a main image annotated onto that canvas with a one-to-one spatial alignment—this means that the image and the canvas are often considered a single entity, and the difference is elided.

The IIIF virtual canvas also allows us a way to uniquely refer to regions of an image that is not tied to a specific derivative or representation of that image. While part of the core specification, in cultural heritage applications this capacity is typically used only when providing transcriptions or displaying OCR. However, given the kind of results that are produced by ML processes, this abstraction of the 2D spatial canvas and the image allows us to overlay additional metadata in a way that aids in human visualization.

[25] https://iiif.io/api/presentation/2.1/

Additional Outcomes and Learnings

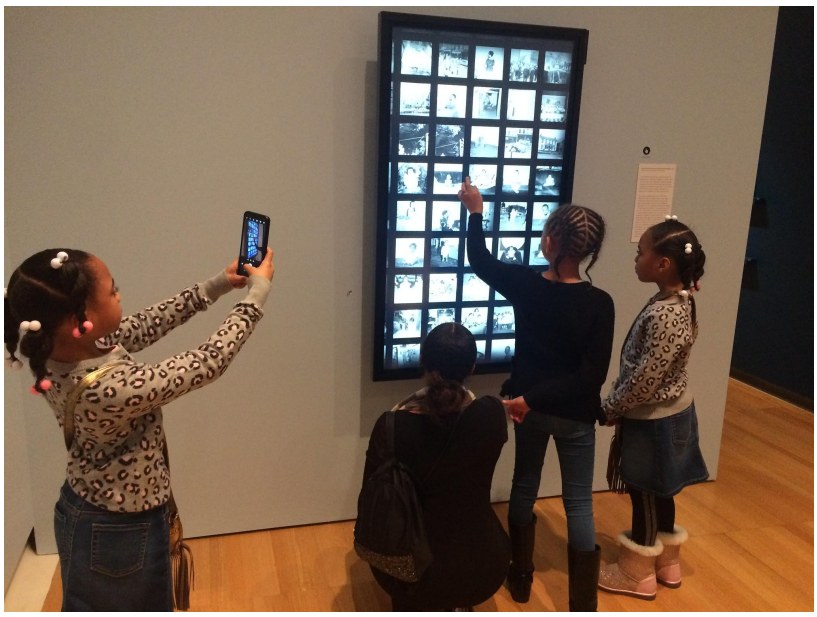
In addition to the data and machine learning workflows we produced for the Teenie Harris Archive at the Carnegie Museum of Art, and our increased understanding about the limitations of IIIF for ML-based archive analysis, there were two other additional, unplanned outcomes of our investigation, discussed below:

1. The development of a public-facing interactive installation for the Carnegie Museum, which was adapted from the annotation interface that we had originally developed for the Museum's archivists, and permanently installed in the Museum; and
2. A technique for bootstrapping the analysis of monochrome photographs using artificial colorization.

Annotation Interface and Public Browsing Station

The many various data we generated are only really useful in the context of an interactive visualization. We created such an application for the Museum's archivists, in order to help them see new patterns in the Teenie Harris Archive and to record authorized face matches where they are discovered. An unintentional outcome of our project is that our application, which demonstrates the utility of the data we generated, was, with small changes, able to be adapted for the general museum-going public.

The Carnegie Museum invited us to permanently install this simplified version of our software, with a 55" touchscreen, in the Museum's new Teenie Harris Room. This opened to the public in late January, 2020. Our interactive visualization is now the digital centerpiece in an otherwise analog room, on view to hundreds of thousands of visitors per year. In the photos below, our installation is shown in use by the great-grandchildren of Teenie Harris himself.



Bootstrapping ML Analysis of Black-and-White Photographic Images with Artificial Colorization

In this section we discuss one of the most original and unexpected breakthroughs of our research project: the use of machine-assisted artificial colorization as a way of bootstrapping further analysis of a grayscale photography archive.

As we discussed earlier, one of our primary strategies for organizing the Teenie Harris photographs has been the use of a convolutional neural network (CNN) to characterize each of the Teenie Harris images with perceptual feature vectors, followed by the use of the UMAP algorithm to reduce the dimensionality of these vectors to a two dimensional arrangement. This allows for an interactive 2D display of the entire image collection, in which the photographs are organized into clusters and neighborhoods according to their visual similarity.

Unfortunately, our initial 2D arrangements of the Harris archive produced this way, just *weren't that cohesive* — especially when subjectively compared to the results of the same workflow when applied to other datasets. On a hunch, we suspected that the problem might be that our neural networks (Inceptionv3 and VGG) had been trained on *color images*, and were having a difficult time making sense of Teenie Harris's grayscale photographs. To "bootstrap" our neural networks, we decided to try colorizing the Teenie Harris images using Jason Antic's *DeOldify* — a recently released, deep-learning based system for image colorization[26] — and then feeding the artificially colorized images into the CNN for re-analysis.

[26] https://github.com/jantic/DeOldify

In the above images, Teenie Harris image #15974 is shown in its original back-and-white form, at left, and at right, artificially colorized by the DeOldify algorithm. The colorized versions, while not intended for public display, produce better-organized arrangements when analyzed by a neural network.

DeOldify provides coarse estimates of color information where none otherwise exists. In short, it makes the sky blue, makes the grass green, and makes human skin ...more-or-less pinkish-brown. The use of DeOldify in this intermediate analysis stage *dramatically* improved the accuracy and legibility of the way in which their corresponding originals could be grouped and clustered. We believe this is a novel technique that has not been used before, and which has broad applicability to the visual analysis of grayscale imagery. Technically, we believe this technique works because we have taken the channelwise mean of each pixel's data, and adapted it to better match the CNN's "correct" (expected) input distribution across channels.

While the use of artificial colorization can lead to dramatic improvements in subsequent image analysis, the colorized images themselves are not intended for public display or release — both because of their divergence from Harris's artistic intention, and because of the fraught social and ethical issues involved in "colorizing" the subjects of his photographs.

Conclusion

With support from the NEH, we have implemented and developed a wide range of workflows that employ machine learning and computer vision to help annotate a large, black-and-white historic photography archive. We have discussed what kinds of metadata can result from that effort, and how IIIF can be used, and where it cannot, as part of an archival analysis project using machine learning. We have discussed how the results of a machine learning workflow can be translated back into the daily practice of cultural heritage archival work.

As an unintentional outcome of our investigation, we developed a public-facing interactive installation for the Carnegie Museum, which has been permanently installed in the Museum. This visualization uses the metadata we computed using machine learning, in order to help the public understand the contents of the Teenie Harris Archive dataset.

Finally, using ML-powered artificial colorization, we developed a technique for improving the analysis of monochrome photographs using convolutional neural nets trained on color datasets.

Acknowledgements

This project was developed at the CMU Frank-Ratchye STUDIO for Creative Inquiry using openFrameworks, Processing, ML4A, and ml5.js, and made possible by support from the National Endowment for the Humanities (Award HAA-256249-17, #1080397); the Carnegie Museum of Art Teenie Harris Archive, the J. Paul Getty Trust, and nVidia Corporation. Developed by Golan Levin and David Newbury (principal investigators); Zaria Howard, Kyle McDonald, Gene Kogan (machine learning); Lingdong Huang (interactive display); Oscar Dadfar, Luca Damasco, Cassie Scheirer (data preparation); Olivia Lynn (additional software). Additional thanks to Dominique Luster, Charlene Foggie-Barnett, Louise Lippincott, Caroline Record, Samantha Ticknor, and the Innovation Studio at the Carnegie Museum of Art; and to Thomas Hughes, Linda Hager, Carol Hernandez, and Bill Rodgers at the CMU Frank-Ratchye STUDIO for Creative Inquiry.

Contributors:

- Golan Levin (CMU), Primary Contact
- David Newbury (J. Paul Getty Museum)
- Zaria Howard (CMU)
- Kyle McDonald
- Gene Kogan
- Lingdong Huang (CMU)
- Oscar Dadfar (CMU)
- Olivia Lynn (CMU)
- Cassie Scheirer (CMU)

- Caroline Record (CMOA)
- Dominique Luster (CMOA)
- Charlene Foggie-Barnett (CMOA)
- Louise Lippincott (CMOA)
- Samantha Ticknor (CMOA)

Institutional Sponsors:

- The National Endowment for the Humanities
- The Frank-Ratchye STUDIO for Creative Inquiry at Carnegie Mellon University
- The Teenie Harris Archive at the Carnegie Museum of Art
- The Innovation Studio at the Carnegie Museum of Art
- nVidia Corporation

Additional Thanks:

- Thomas Hughes (CMU)
- Linda Hager (CMU)
- Bill Rodgers (CMU)
- Carol Hernandez (CMU)
- Aman Tiwari (CMU)
- Omer Shapira (nVidia)