BuWvtJFxh3wy

# Tracking and tracing audiovisual reuse: Introducing the Video Reuse Detector

## XXX

## XXX

## XXX

[ ©. Published by De Gruyter in cooperation with the University of Luxembourg Centre for Contemporary and Digital History. This is an Open Access article distributed under the terms of the [Creative Commons Attribution License CC-BY](https://creativecommons.org/licenses/by/4.0/)

The reuse and reappropriation of audiovisual content have been a recurring topic of research in the humanities, not least in studies of remix cultures. An open question that remains, however, is how artificial intelligence and machine learning may help scholars study the reuse of audiovisual heritage. In this article, we introduce the Video Reuse Detector (VRD) – a methodological toolkit for identifying visual similarities in audiovisual archives with the help of machine learning. Designed to assist in the study of the “social life” and “cultural biographies” (*Kopytoff 1986*, *Appadurai 1986*) of video clips, the VRD helps explore how the meaning of historic footage changes when it circulates and is recycled/cross-referenced in video productions through time. The toolkit uses machine learning techniques (specifically, convolutional neural networks) combined with tools for performing similarity searches (specifically, the Faiss library) to detect copies in audiovisual archives. It also assembles a series of tools for trimming and preparing datasets and filtering/visualizing matching results (such as introducing similarity thresholds, filtering based on sequential matches of frames, and visually viewing the final matching results). Inspired by the “visual turn” in digital history and digital humanities research, the article introduces and exemplifies the basic logic and rationale behind the VRD, and discusses how the digitization of audiovisual archives opens new ways of exploring the reuse of historic moving images.

Digital methods, Machine learning, Cultural reuse, Video archives

## Introduction

On February 28, 2023 the Swedish YouTuber [RosaMannen](https://www.youtube.com/user/RosaMannen) — the Pink Man — uploaded an [episode](https://www.youtube.com/watch?v=arIukjQWxuU&t=783s) from the TV documentary series *Sweden, A Long Time Ago* on YouTube. Originally broadcasted on Swedish public television in 1997, the clip was 18 minutes long and surveyed the history of Swedish handicrafts, ranging from knitting to the tying of brooms and the production of metal nails. Unspectacular in its execution, the episode featured black-and-white footage of farmers preparing wool, cutting grass, and craftsmen building wooden ladders. Since 2006, RosaMannen (also known as Daniel Johansson) has uploaded tens of thousands of similar TV clips on his YouTube channel, which now provides an impressive overview of Swedish television history. Alongside episodes from the TV series *Sweden, A Long Time Ago* — which for example also includes episodes on [industrialization](https://www.youtube.com/watch?v=MettI_h2Oiw) and [school children](https://www.youtube.com/watch?v=yT5ffg_mkQY&t=9s) — his channel features an eclectic mix of cooking shows from the early 1980s and 90s, children’s shows, televised concerts by popular Scandinavian artists, investigative journalism series, historic news reports, and much, much more. These videos have accumulated millions of views and roughly 40 000 subscribers now follow RosaMannen’s channel on a day-to-day basis. Indeed, Johansson has become something of a national video celebrity, whose persistent commitment to digitizing either donated or purchased home-recorded VHS tapes has made a vast number of Swedish television shows available to the wider public.

RosaMannen’s way of making Swedish TV history available on YouTube is illustrative of amateur archivist practices and a flourishing online video culture. His uploading of content is, however, far from legal. RosaMannen’s logo features a man in a pink suit wearing a Guy Fawkes mask yet despite this anti-establishment protest, Swedish Television has refrained from posting take-down requests. During the past decades, a massive body of audiovisual heritage has been digitized and made available online – not just by television enthusiasts such as Johansson, but by a wide range of cultural heritage institutions. This has radically transformed public access to cultural heritage and fostered new ways of engaging with the audiovisual past. By increasing the visibililty of historic video content and allowing it to circulate online, audiovisual memories are constantly revived, rewritten, and renegotiated. This does not least include breathing new life into seemingly mundane and low-key video clips – such as the episode on Swedish handicrafts – which otherwise likely would have drifted far into oblivion. Accompanied by ample user commentaries that interweave nostalgia and personal memories with discussions about nationalism, economics, and politics, the videos on RosaMannen’s YouTube channel are an excellent example of how digital technologies help remediate cultural memories.

Importantly, the digitization of audiovisual heritage has also opened up new ways of studying video archives at scale, as moving image collections can now be explored through computational means. In similar ways as the field of literature has expanded to engage with “distant reading” in the large-scale analysis of text (*Moretti 2013*), scholars have highlighted the potential in engaging with “distant viewing” in the context of audiovisual scholarship (1971321/3LPY6AMR) (*Arnold, Tilton 2019*) (*Arnold, Tilton 2019*). For instance, previous studies have explored audiovisual content using algorithmic object and facial recognition techniques (ibid.), geographic mapping technologies (*Taurino, Boni 2018*), and automatic color extraction and analysis (*Barbara Flueckiger 2017*). This growing body of research has inspired researchers to speak of a “visual turn” in the digital humanities, which takes advantage of how “computer vision techniques offer us a macroscopic perspective on thousands of images” (*Wevers, Smits 2019*, pp.24).

In this article, we explore how audiovisual reuse can be tracked and traced through algorithmic means. Starting from a selection of television documentary series that are available on RosaMannen’s YouTube channel, we investigate how digital tools can assist in studying video reuse in historic compilation documentaries. In particular, we set out to explore how footage in some of RosaMannen’s digitized television series can be traced back to the so-called SF archive, which contains some of the earliest recordings in Swedish film history. The archive of the Swedish Film Industry (SF) includes roughly 5,500 newsreels and film fragments from 1897 to the early 1960s. Most of the oldest footage was filmed by cinematographers from Swedish Biograph, a company that in 1919 merged into SF (*Olsson 2022*). In 1964, the SF-archive was bought by Radio Sweden (also responsible for television) – several short films from SF were purchased later – and ever since footage from the archive has been featured in thousands of TV programs. Around the millennium, the entire SF-archive was digitized in a collaborative effort between Radio Sweden and the Swedish National Archive of Recorded Sound and Moving Images. In our experiment, we work with roughly 1400 of the oldest digitized clips found within the archive (from the silent period of film history) to explore how reuse has taken place between the archive and the selected television series. Four different TV series from the 1980s, 90s and early 2000s were chosen, based on RosaMannen’s digitization efforts: (1.) *Förväntningarnas tid* (The Time of Expectations) from 1985, a series in four hour-long episodes; (2.) *Guldkorn* (Gold Nuggets) from the SF-archive, six hour-long episodes from 1985 and six from 2002; (3.) *Sverige för länge sedan* (Sweden, A Long Time Ago), a TV series from 1984 in ten episodes, each around twenty minutes long; and (4.) *Hundra svenska år* (A Hundred Swedish Years), eight hour-long episodes from 1999. Since all these TV series focus on Swedish 20th-century history, we knew they would likely include reused content from the SF-archive. However, we were unaware of the scale and precise details of what such reuse would look like.

To assist us in unraveling reuse between the SF-archive and the selected TV series, we introduce the Video Reuse Detector, or VRD – a toolkit for detecting visual similarities in audiovisual datasets. The VRD opens up for asking questions such as: How is video content recycled, cross-referenced, and reused in video archives throughout history? And in what ways can the “social life” and “cultural biographies” (*Kopytoff 1986*, *Appadurai 1986*) of audiovisual content be traced? These questions aim to follow the histories and career paths of individual video clips and explore how their cultural value and meaning transforms as they circulate in space and time. As Wolfgang Ernst notes, automated techniques for identifying and processing cultural content provide a unique opportunity to side-step traditional metadata (i.e., textual/semantic descriptions of cultural content) to instead perform searches *within* content itself (*Ernst 2021*). Thereby, they also introduce a radically new way of navigating and interacting with historic archives.

In the following text, we first provide an overview of previous research concerning cultural reuse. We also discuss uses of audiovisual analysis tools within digital history and the digital humanities. This overview shows that even though scholars have started using AI to analyze and navigate audiovisual archives, there is a lack of open-source toolkits that are customized for studying video reuse. Next, we introduce the Video Reuse Detector; an audiovisual content identification toolkit developed at Humlab, Umeå University. Using machine learning models (convolutional neural nets) and indexation libraries (Faiss), the VRD allows for studying how audiovisual content is reused, remixed, and re-appropriated within a given archive. Here, we make use of the Journal of Digital History’s notebook setup to illustrate how the VRD functions and apply the toolkit to two demonstration videos: [The Eagle Has Landed: The Flight of Apollo 11](https://archive.org/details/gov.archives.arc.45017) which contains original footage from the first moon landing in July 1969 and was released by the U.S. National Aeronautics and Space Administration (NASA) in the same year, as well as [The Moon and Us](https://archive.org/details/journey-through-the-solar-system-episode-06-the-moon-us), which consists of Episode 6 from the documentary series *Journey Through the Solar System* that was also produced by NASA and contains footage from Apollo 11 mission. The data and code needed to run this toolkit demo on any computer are openly available on Github, making it possible for anyone to reproduce the demo and test/interact with the toolkit by themselves.

After this demonstration, we show how the VRD can also help in the analysis of larger audiovisual datasets, namely the aforementioned SF-archive and TV series. In this case, however, the original datasets used could not be made openly available online for copyright and practical reasons – apart from the TV series that are already available on RosaMannen’s YouTube channel. Hence, the second part of the article is closed and non-interactive, yet aimed to illustrate the broader potential of applying the VRD to larger sets of source materials. Our demonstration of the VRD toolkit is thus simultaneously an attempt to introduce a method for using machine learning to study video reuse – and an attempt to explore how reuse took place between the SF archive and historic, Swedish television documentaries in the 80s, 90s and early 2000s. Here, we also reflect upon the double remediation taking place as RosaMannen later picks up on and reuses the same video content on YouTube.

### Studying reuse with neural nets

The reuse, remixing, and reappropriation of cultural content – such as the production of compilation documentaries or the circulation of video content online – has been a recurring topic of investigation in the humanities. Central to the field is exploring how cultural memory and identities are (re)shaped through the circulation of cultural content, including things such as images, texts, and films. As David Jay Bolter and Richard Grusin note, cultural memory is fundamentally built on the “remediation” and repurposing of already existing memory-matters (*Bolter, Grusin 1999*). Through acts of reuse, memories become created, stabilized, and consolidated; and “just as there is no cultural memory prior to mediation there is no mediation without remediation: all representations of the past draw on available media technologies, on existent media products, on patterns of representation and medial aesthetics” ((Erll and Rigney 2009, p. 4)). To borrow from Aleida Assmann: the medium and media of cultural memories therefore matter (*Assmann 1996*) and by studying how specific cultural objects (letters, photographs, moving images, etc.) have been copied and reused, we can gain important clues regarding the transformation of cultural ideas and identities over time. This is true regardless if we are interested in the history of Shakespeare’s manuscripts, the afterlife of home-recorded VHS tapes on YouTube or the circulation of viral TikTok memes.

The historic study of cultural reuse has also taken many forms – stretching from philology in literature (*Greenblatt 1988*), to the study of remix and sampling in popular culture (*Navas 2012*, *Lessig 2008*). In the particular case of television history, many have pointed out the special role of TV in shaping cultural memory, creating nostalgia, and reproducing historic narratives (*Holdsworth 2011*, *Wales 2008*, *Macdonald 2006*). Yet the research field is also faced with methodological challenges. To this date, scholars of cultural reuse have mainly had to follow traces in individual documents (such as references to previous works) or rely on textual metadata. For example, studying how a particular photograph has been reused through time may have involved searching through archives using metadata keywords, and manually studying interesting finds. This is highly time-consuming and implies relying on textual descriptions that may be of poor quality and level of detail. As a result of digitization, however, it becomes possible to ‘look into’ and search through the actual contents in archives at a scale that has not been possible before. Drawing inspiration from fields such as digital forensics (*Kirschenbaum 2008*), we can begin to use computational techniques to investigate the origins and authenticity of cultural objects and search for visual similarities in large archives.

To this date, automatic content recognition techniques have been amply explored and developed in various computer science fields such as machine vision and information processing/retrieval. Early approaches for detecting image reuse included algorithmic techniques for finding geometric shapes in images, watermarking which drew on stenographic techniques to encode messages in images to make them recognizable, as well as hashing and feature detection technologies like the ORB, or [Oriented FAST and Rotated BRIEF](https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html), which finds visual similarities by identifying pixel intensities and key points. A problem with many of these early techniques, however, was that they were slow in processing large amounts of data. Furthermore, they were often unable to detect visual similarities if content had been modified. While moving images that have been copied to the point – minute by minute, frame by frame – are often fairly easy to recognize, partial reuses of video content (where short segments of videos are copied), or instances where videos have been heavily remixed or manipulated (for example by changing their color, sharpness, composition or playback speed) presents a more difficult task. To face this problem, the field of machine vision and image retrieval has increasingly turned to artificial intelligence and machine learning, where techniques such as Convolutional Neural Networks (or CNNs) provide radically new ways of searching for visual similarities in datasets.

This is also true in commercial contexts, where the automatic detection of copied cultural content is crucial to efforts to enforce copyright and restrict the circulation of pirated content online. In 2018, for instance, Google claimed that it had invested more than $100 million in building Content ID; an automated system for detecting copyright abuse that scans and monitors reuse in *all* videos that are uploaded on YouTube (*Google 2018*). Google claims that Content ID is responsible for handling more than 98 percent of the copyright disputes taking place on YouTube, meaning that the technology plays a key role in controlling how content enters and exits one of the world’s largest websites. Elsewhere, businesses like Audible Magic have also developed advanced systems for monitoring how copyright-protected content circulates online. Specialized in the real-time identification and classification of recorded speech, videos, and music, Audible Magic claims to have the capacity to identify how 25 million “media assets” stemming from 1000 video suppliers and 140,000 record labels appear in television broadcasts, on streaming platforms, archived TV, and the motion pictures worldwide (Audible Magic, 2020). Would it be possible to adopt similar techniques as those developed in computer science and the rights industry to study cultural reuse?

Recently, scholars in the humanities and social sciences have begun to show how machine learning can be applied to visual archives to answer historic research questions. While not immediately concerned with the study of cultural reuse, such research has illustrated how artificial intelligence can assist in navigating and exploring digital archives in radically new ways. For instance, scholars have used artificial intelligence to identify scenes in collections of historic press photos (*Wevers, Vriend, De Bruin 2022*) and to explore gender representations in millions of historic advertisements (*Wevers, Melvin, Smits 2020*), as well as nearly 3400 issues of Times Magazine (*Jofre, Cole, Berardi, Bennett, Reale 2020*). Others have used machine learning to conduct portrait studies in historic photographic datasets (*Arnold, Tilton, Wigard 2022*) and explored how iconic photographs are (and aren’t) memorized with the help of AI (*Saleh, van Noord 2022*).

A number of projects have also developed methodological toolkits for exploring audiovisual content with the help of machine learning – a source material that is often more complex to work with than still images because of its temporal dimension and the large size of datasets. Consider, for instance, that digital videos generally contain 24-30 frames or images per second, which means that 10 minutes of video may contain 14 400-18 000 images alone. This poses problems in terms of access to computational power. In the United States, however, the [Media Ecology Project](https://mediaecology.dartmouth.edu/wp/) at Dartmouth College has prototyped a machine vision search engine for identifying objects and actions in videos, and at the University of Richmond, the [Distant Viewing Lab](https://distantviewing.org/) has released a Python package for analyzing visual culture on a large scale, including features like facial recognition. In Germany, the project [Visual Information Retrieval in Video Archives](https://projects.tib.eu/en/viva/projekt/) has also been established to support media archives by developing a ‘videomining software’ that can for example be used to classify events (*Müller-Budack, Springstein, Hakimov, Mrutzek, Ewerth 2021*), person relations (*Müller-Budack, Pustu-Iren, Diering, Ewerth 2018*) and video scenes based on geolocation (*Müller-Budack, Pustu-Iren, Ewerth 2018*).

Similarily, [The Sensory Moving Image Archive](https://sensorymovingimagearchive.humanities.uva.nl) project in the Netherlands has experimented with stylometric film analysis (i.e. exploring characteristic stylistic features of films and filmmakers) and released a so-called [Nearest Neighbour Browser](https://isis-data.science.uva.nl/nanne/SEMIA/#662692_1) that allows for exploring visual similarities in datasets (although without using specific images as input). The toolkit [PixPlot](https://github.com/YaleDHLab/pix-plot), developed by scholars at the Digital Humanities lab at Yale, does something similar, as it allows for two-dimensional clustering and visualization of visual similarities in tens of thousands of images. Another example is the software Snoop, which has been developed by the French Institut National de l’audiovisuel (Ina) in collaboration with the French National Institute for Research in Digital Science and Technology (Inria) since 2003. Snoop, which previously went under the name Signatur, is a visual search engine that applies the open-source machine learning framework PyTorch to find visual similarities in datasets (*Delfieu, Lombardo, Joly, Leroy, Busson 2022*). The toolkit has been used to explore and enrich visual archives by researchers connected to Ina/Inria, as well as cultural heritage institutions such as Musée d’Orsay and the Bibliothèque Nationale de France. It has also been used in citizen science projects such as [PlantNet](https://plantnet.org/en/), where it assists in identifying plants in photographic images. However, Snoop is a [proprietary software](https://hal.science/hal-02096036v1) that is not openly available for use.

In the toolkits mentioned above, machine learning models have mainly been used to produce semantic metadata and textual interpretations of what appears in images (including scene, object, or facial recognition), or to find *general* visual similarities in datasets. For example, the toolkits are often geared towards finding examples of when say, astronauts, appear in images so that several depictions of astronauts can quickly be retrieved and studied. In our case, however, we are not interested in finding examples of how astronauts appear in datasets *in general*. Rather, we want to explore how a *specific* depiction of a *specific* astronaut appears in a dataset - say footage of Neil Armstrong taking his first steps on the moon. And toolkits that are specialized in finding these kinds of audiovisual reuse are less common to find.

## Video Reuse Detector

In response to the lack of open-source software for audiovisual copy detection, we introduce the VRD – a methodological toolkit for identifying visual similarities in audiovisual archives with the help of machine learning. The VRD is meant to help archivists and humanistic scholars study video reuse and was originally developed within the research project XXX, funded by XXX. Its main developer is XXX, with assistance from XXX and XXX. The toolkit handles all video formats supported by the FFmpeg framework and is built to be used in Jupyter Notebooks. It is free to use and download and the source code is openly available on Github (link to Github page omitted in this article version to maintain anonymity).

In what follows, we demonstrate how the toolkit functions by applying the VRD to two videos that are openly available on [Archive.org](https://archive.org/). The first video, entitled [The Eagle Has Landed: The Flight of Apollo 11](https://archive.org/details/gov.archives.arc.45017), contains footage from the first moon landing in July 1969 and was released by the U.S. National Aeronautics and Space Administration (NASA) in the same year. The second video, entitled [The Moon and Us](https://archive.org/details/journey-through-the-solar-system-episode-06-the-moon-us), consists of Episode 6 from the documentary series *Journey Through the Solar System* which was also produced by NASA. First released in 1983, the series contains footage from various Apollo missions, including Apollo 11. Both clips are roughly 30 minutes long and are used in compliance with their respective licenses. Before we used the videos in this demo, we compressed them with HEVC to stay within Github’s allowed file size of 100 MB/file.

When assembling the VRD, the following rationale has guided our work:

* All applied software solutions are open source. In other words, they are licensed to be openly available for anyone to use, study, change, and distribute without cost.
* All applied software solutions can be exchanged at any time. This is ensured by the modularity of the VRD’s Jupyter Notebook and its associated Github page which is open for anyone to tweak and adjust.
* No data is exchanged with commercial actors. Aside from sharing information with Jupyter Notebook and local storage systems chosen by the user, the VRD does not transmit any data to third-party actors (including those who developed parts of its code).

The VRD toolkit is designed to process data in seven main steps which are introduced in the following sub-sections. Technical details and a step-by-step tutorial that explains how the toolkit works are placed in the article’s hermeneutics layer. In the article’s narrative layer, we present a textual description of the toolkit’s main functionality principles. As previously stated, it is possible to download and run a local copy of the demo section by visiting the project’s Github page. Readers who are less interested in methods and the technical workings of the VRD are advised to jump to the section [Finding reuse in the SF archive](#finding-reuse-in-the-sf-archive).

### Step 1. Pre-process videos

To begin with, the VRD includes tools for dividing audiovisual content into still frames. Digital videos generally contain 24-30 frames per second and the VRD is instructed to extract one frame per second of video by default. These still images constitute the VRD’s main object of analysis. Since our research team was originally interested in performing visual similarity searches, the current version of the VRD does not analyze sound. This means that the tool cannot detect if the audio track of a video file has been reused and reappears somewhere else. The rationale for this is that reused video is often dubbed/overlayed with different audio, meaning that the audio might not match even in cases where the video is reused. Adding audio features to the VRD has therefore not been a priority at this stage.

START HERMENEUTICS

We begin by importing a series of necessary modules.

END HERMENEUTICS

START HERMENEUTICS

We also install ffmpeg since it is a prerequisite for ffmpeg-python.

END HERMENEUTICS

START HERMENEUTICS

We start a new project and give it a name. We also inform the VRD where to locate the video files we want to work with and choose to apply the neural network EfficientNetB4 from the Keras API. We decide to export vectors/fingerprints from a layer called Block7b\_se\_squeeze in EfficientNetB4.

END HERMENEUTICS

START HERMENEUTICS

We extract one frame per second from each video file. These frames are saved in the project directory, under the frames subdirectory. As can be seen in the example below, this meant that 3419 frames were extracted from the source material.

END HERMENEUTICS

### Step 2. Produce fingerprints

Second, the VRD uses a so-called Convolutional Neural Network – or CNN – to extract the key visual features found in each frame. We call these extracted visual features “fingerprints.” CNNs constitute a common technique for studying visual imagery with the help of artificial intelligence. Modeled to imitate the connectivity pattern of neurons in the visual cortex of animals, neural networks are used in areas such as facial recognition, medical imaging, and autonomous driving. We decided to work with CNNs after doing multiple tests with more traditional tools for visual content recognition, including image hashing and video fingerprinting with the help of [ORB](https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html) (Oriented FAST and rotated BRIEF) – a method for extracting and comparing visual features in images. CNNs quickly outperformed the ORB technology’s ways of identifying visual similarities in video content, however, both in terms of accuracy and processing speed.

It is beyond the scope of this article to explain how CNNs work in depth (for more information on the topic, one place to start is [Wikipedia](https://en.wikipedia.org/wiki/Convolutional_neural_network)), but we make a couple of remarks regarding the technology’s basic technical structure. While the detailed technical workings of individual CNNs differ, neural networks are broadly designed according to multiple layers of analysis and abstraction. Each layer in a CNN will process an input and produce an output, which is passed on to the next layer. For instance, one layer in a CNN may observe how pixels are spatially arranged in an image and search for areas with high contrast between nearby pixels (a good marker for what is visually unique in a picture), while another layer might focus on reducing what information is stored about pixel contrasts (instructing the model to “forget” all areas in a picture with a lower pixel contrast than a given value, for example). In this way, the CNN produces a successively smaller and hopefully more precise map of the analyzed image. Somewhere before the final layer of a CNN is reached, the network will produce a highly compressed interpretation of the key visual characteristics of images. It is then common for the remaining layers in a CNN to classify what appears in the image, for instance by recognizing faces and objects.

In our case, the VRD applies a CNN to process individual frames but stops when a compressed yet sufficiently complex interpretation of the key visual features of an image has been produced. Again, we call these compressed interpretations fingerprints and the VRD will use them to find patterns of similarity across videos. In more detail, the VRD will export data in the form of vectors from a given layer in a CNN. These vectors mirror the visual characteristics found in the original frames. A vector is a mathematical quantity that discloses the magnitude (or length) and direction of a geometric object. In other words, we can say that fingerprints are mathematical abstractions that carry information about what is visually unique in each frame. What is good about fingerprints is that their detailed yet highly abstracted information can be used to recognize visual similarities even if a video has been modified or distorted. For instance, frames can be recognized as similar even if someone has adjusted their color, resolution, or composition. This is useful when studying reuse and remix practices.

When the VRD extracts its vectors/fingerprints from a given intermediate layer in a CNN – something that is otherwise done to visualize the learning process of a CNN, for example – the remaining layers of a neural net are not evaluated (more about this in the [Keras FAQ](https://keras.io/getting_started/faq/#how-can-i-obtain-the-output-of-an-intermediate-layer-feature-extraction)). We partly do this to save time/computing power, and partly because we are not interested in classifying the content in images. Instead, we essentially use a CNN to compress and vectorize video frames. It is important to understand that this is not how CNNs are normally used. Therefore, we also know relatively little about which CNN and extraction layer are best suited for the task. Commonly, the performance of CNNs are evaluated based on tests such as the [ImageNet Large Scale Visual Recognition Challenge](https://www.image-net.org/challenges/LSVRC/), where a neural network’s capacity to recognize objects/faces in images is compared against human beings’s ways of performing the same task. Tests such as the ImageNet Challenge are not designed to evaluate a CNNs ability to compress and vectorize frames, however. When looking at previous evaluations of CNNs and deciding what neural network to use (see [Keras documentation](https://keras.io/api/applications/)), we have therefore mainly searched for CNNs that are relatively new and perform well in overall tests.

The current version of VRD is based on [TensorFlow2](https://www.tensorflow.org/), which includes the open-source Keras API. TensorFlow2 is an open-source machine learning platform and software library for working with machine learning and artificial intelligence. It was originally developed by Google and can be used for multiple purposes, including the training and inference of deep neural networks such as CNNs. The [Keras API](https://keras.io/about/) is a deep learning API that runs on top of the TensorFlow platform. Keras functions as an interface for solving machine learning problems and provides building blocks for using machine learning methods. We use version 2.11.0 of the Keras API which makes 11 pre-trained convolutional neural networks available for use. All of these networks are open source and can easily be applied in the VRD. We mainly decided to work with pre-trained neural nets since it was unfeasible for our small research team to train a new network from scratch. Furthermore, we found that many of the neural nets in the Keras API succeeded in producing vectors that were compressed yet detailed enough to study video reuse, without needing to be re-trained or fine-tuned. Hence, we decided to work with the networks directly in their original form. Re-training the networks on relevant datasets would likely further improve the performance of the VRD, however.

Initially, we mainly worked with the neural network [ResNet50](https://keras.io/api/applications/resnet/) (see Snickars et.al. 2023) but later switched to a network called [EfficientNetB4](https://keras.io/api/applications/efficientnet/), which is newer and performs better in accuracy tests. EfficientNet was first released by engineers at Google Research’s Brain Team in 2019 (*Tan, Le 2020*), and at the time of this article’s writing, it existed in seven different versions (see [Keras documentation](https://keras.io/api/applications/efficientnet/)). We have found that version B4 – which lies in the mid-range in the tradeoff between accuracy and speed among the EfficientNet versions – works well to identify video reuse. The current version of the VRD applies EfficientNetB4 *as is* (that is, without any re-training) and extracts its fingerprints from a layer called ‘block7b\_se\_squeeze’. This layer is found towards the end of the network model (at the time of this article’s writing, layer 463 out of 476). When deciding where to extract our fingerprint vectors, we wanted to find a layer that contained complex interpretations of the visual features in frames, yet produced vectors that were sufficiently compressed to work with large datasets. In addition, we wanted to find a layer where the neural network had not started to apply its object classification too strictly. In our tests, Block7b\_se\_squeeze appeared to live up to these qualifications. Importantly, however, we recommend exploring other layers (and CNNs) when using the VRD, as we have not performed any comprehensive performance tests of all CNNs in the Keras API and their respective layers.

START HERMENEUTICS

We use EfficientNetB4 to create a fingerprint for each extracted frame and use these to populate the fingerprint database. The database is saved as ‘database\_file’ in the project directory. Note that it is important to delete any pre-existing databases and start over if another CNN is used. This is done by specifying force\_recreate=True.

END HERMENEUTICS

### Step 3. Calculate similarity neighbours

After saving extracted fingerprints in a database, the VRD applies its third step, where the [Faiss library](https://faiss.ai/) is used to calculate the closest similarity neighbors for each fingerprint. Faiss is an open-source software library that specializes in large-scale similarity searches. First released by Facebook AI Research in 2018, it efficiently clusters dense vectors and conducts large-scale similarity searches. For instance, Faiss can be run on Graphical Processing Units (GPUs) which provides significant advantages in terms of speed. As an open [Faiss manual](https://www.pinecone.io/learn/faiss-tutorial/) explains, Faiss will index a given set of vectors and then use another vector (called the query vector) to search for the most similar vectors within the index. Faiss allows for determining which vectors are similar by measuring the Euclidean distance between *all* given points within an index using an index called IndexFlatL2. FlatL2 performs an *exhaustive* search and is very accurate in its evaluation of vector similarity, but slow since it matches/compares *every* point in the index (in our case, fingerprints) against the other, one by one. To speed up this process, the Faiss index contains several methods for optimizing similarity searches, although these will always be implemented at the cost of accuracy.

For instance, it is possible to optimize the speed of a Faiss similarity search by partitioning the index (i.e., limiting the search scope to *approximate* similarity, rather than calculating an absolute similarity) or by way of quantization, which involves compressing the size of vectors (more about this in the following [Faiss tutorial](https://www.pinecone.io/learn/faiss-tutorial/)). When the Faiss index has determined (or approximated) the similarities found within an index, it will output a “distance metric” to each compared set of vectors. The distance metric is a value that indicates how closely related their features are, according to Faiss. A low distance metric value (or short distance) indicates high similarity and a high distance metric value (or long distance) indicates low similarity. The distance metric 0.0 represents the absolute closest similarity Faiss can ascribe to two compared vectors and essentially corresponds to the distance that a vector would have to itself (i.e., an absolute match).

While Faiss can be used for any type of similarity search, the VRD uses it to identify visual similarities between frame fingerprints. In particular, the VRD will apply IndexFlatL2 to perform an exhaustive search and compare all fingerprints against each other, without optimizing the similarity search. While this is costly in terms of speed/processing power, it allows the VRD to later find sequential matches in the analyzed videos - a feature that is central to how the toolkit works. VRD users can overwrite the use of IndexFlatL2 and instead use the Faiss library’s optimization methods. However, this implies that the VRD’s current structure for outputting final matching results in the form of sequential matches will be lost.

To save time/memory and minimize the display of poor similarity matches, the Faiss index comes with a default setting that only shows the 100 closest neighbors for each analyzed vector. While it is generally desired to limit the number of neighbors shown for each vector, this threshold comes with drawbacks because of how the VRD is built. More specifically, it is important to note that the threshold is applied before the VRD runs another filter that removes all matches from the same video. This can cause problems if a video contains a lot of still images or slow visual movements since long sequences of frames from the same video could then be given a very low distance metric. In such cases, a frame’s 100 closest neighbors may be occupied by lots of frames from the same video, while other relevant matching results are “pushed” out of the top 100 list. When all matches from the same video are filtered out from the top 100 similarity neighbors, important matches could thus be lost. While it would be preferable to filter matches from the same video *before* distance metrics are calculated and exported, the Faiss library does unfortunately not support this feature at the moment (a problem that has also been noted by [others on Github](https://github.com/facebookresearch/faiss/issues/40)). It is, however, possible to adjust the 100 nearest neighbor threshold to reduce the risk of filtering out interesting matching results. This is done in the VRD, as the threshold is increased from 100 to 250 by default.

The distance metrics produced by the Faiss library constitute a core element of the VRDs evaluation of visual similarity, although it is important to note one final thing, namely that these metrics are dynamic and will change/fluctuate depending on which dataset is processed. For instance, the quality of the source material and the number of images/frames in the dataset will affect how distance metrics are distributed. Likewise, the distribution of distance metrics is highly affected by which neural network and neural network layer is used. This means that there is no absolute value or threshold that indicates what distance metric value corresponds to a “correct” or “actual” instance of reuse. Instead, any use of the VRD will always involve manually exploring how each project’s unique distance metric values correspond to actual visual likenesses in the source material.

START HERMENEUTICS

We instruct Faiss to output the 250 closest neigbors for each fingerprint. These will be saved in the “neighbour\_batch” subdirectory in the project directory. The VRD will apply Faiss IndexFlatL2 as is (i.e. perform an exhaustive search where all fingerprints in the index are compared against each other). To change this setting and make use of the Faiss library’s similarity search optimization, changes have to be made to the source code. Note, however, that doing so will result in VRD’s ways of finding sequential matches being lost.

END HERMENEUTICS

START HERMENEUTICS

We create a Faiss index of the fingerprints and save it as faiss\_index in the project directory. If the index has already been created, the VRD will load the last saved index by default. Note that if any changes are made to the source material (i.e., the videos, frames, or fingerprints) or the selected CNN model it is necessary to recreate the index by passing the argument force\_recreate = True.

END HERMENEUTICS

### Step 4. Analyze distance metric distribution

To determine how distance metric values should be interpreted, it is possible to view a distance metric visualization (histogram) that displays how metrics are distributed within a dataset. The visualization is useful for figuring out where to place a distance metric threshold and filter out irrelevant matching results. If the threshold is set too low, interesting matching results may get lost. If the threshold is set too high, there is a risk of being shown a high number of uninteresting matches.

START HERMENEUTICS

We create a histogram to get an idea of where it might be suitable to place a distance metric threshold. When using the tool, we have found that placing the threshold around the point where the histogram indicates a sharp increase in found neighbors is a good start, although the specifics will be determined by the content and size of the dataset.

END HERMENEUTICS

### Step 5. Filter matching results

In the final step of the VRD’s image processing, it is possible to narrow down the search results with the help of two major filtering features. To begin with, it is possible to implement a general distance metric threshold before the final matching results are shown. For instance, the VRD may be instructed to only show fingerprint neighbors with a distance metric below the value 200. An accurate threshold (again, manual exploration is always necessary here) should greatly reduce the number of incorrect matches.

Second, the VRD includes a feature for applying what we call “sequential filtering.” We define a sequence as an instance where two or more sequential fingerprints (i.e., frames that were extracted one after the other from the original file) from two videos that have been given a distance metric below a specified value. If frame 1-6 in Video X and frame 11-16 in Video Y are each given a distance metric below the threshold 200, for example, this may be defined as a sequential match.

Sequential filtering is used to identify instances when longer chunks of moving images have been reused and we assume that such chunks are more interesting to users than individual matching frames. Furthermore, we have found that sequential matches are generally more indicative of actual reuse than individual matching frames since there is a higher likelihood that actual cases of reuse have been found when at least two frames in a row have been assigned a low distance metric. Sequential filtering is implemented by deciding the minimum sequential length, or duration in seconds/frames. It is also possible to instruct the VRD to ‘tolerate’ that a given number of frames within a sequence deviates from the assigned distance metric threshold using the allow\_skip parameter. Allowing some skipping is recommended, as it i s likely to improve results.  For instance, it allows a sequence to be detected even if scene changes occur at slightly different times in different videos.

Importantly, the VRD’s sequential filtering also includes an option called combine\_overlap. As the name implies, the combine overlap option is a best-effort attempt at combining overlapping sequences into one. Overlapping sequences is an issue that may occur if several seconds in a reused video looks very similar. Examples include reused videos with still footage, or slowly changing scenes such as panoramic overviews of landscapes. If this feature is disabled, it can lead to multiple similar sequences which only differ slightly to be presented in later steps.

For example, let’s imagine that a scene depicting an ocean has been reused in two videos. In the reused scene we see a clear blue sky, a horizon, and moving waves but nothing else is happening in the picture. In such an example, several overlapping sequences may be found relating to the same footage. For instance, the VRD may find that frame 5-10 in Video 1 look very similar to frame 20-25 in Video 2. But the VRD may also find that frame 6-11 in Video 1, and frames 21-26 in Video 2 have a high visual similarity. This creates an overlapping sequence match with a slight time shift (i.e., starting one or two seconds before or after). If such overlapping sequences are not merged into one, users are faced with unneccesseary clutter when looking at the sequential matching results. Hence, our solution is to combine overlapping sequences into one single sequence.

To combine overlapping sequences, we search for sequences that overlap (i.e., contain the same frames), and merge them by starting at the earliest start time and ending at the latest end time. The algorithm can combine multiple different sequences into one, even if any one sequence does not overlap with all others. In the best-case scenario, the result is a single longer sequence that contains frames from two videos that are perfectly aligned. However, in some cases, a cumulative error from merging multiple overlapping sequences can result in a combined sequence that is significantly degraded from the original in the current implementation. This degradation is more likely to appear when the actual overlap has an offset of several seconds. As a consequence, the reported distance metric for the sequence (which is calculated from the mean of the frame-by-frame distance) can become misleadingly large, or even in some cases undefined. While this issue is currently unresolved, we believe that this can be mitigated by performing a final matching step where the sequences are realigned to better match. A preview of such a feature can be viewed by specifying show\_shift=True as an argument to the show\_notebook\_sequence function.

To recapitulate, the VRD performs its similarity search by converting videos to still frames and still frames to fingerprints. A similarity search is then performed on the fingerprints, producing for each frame a list of similar frames. Alternatively, one could describe the process as a matter of compressing, converting, and abstracting ‘raw’ audiovisual content to frames, frames to vectors, and vectors to distance metrics. Furthermore, the VRD contains tools for narrowing down the search results, including features for implementing distance metric thresholds, filtering out matching results from the same video, and applying sequential filtering, that is looking for instances when two sequences or more in a row have been assigned a low distance metric value.

START HERMENEUTICS

We configure some filters to narrow down the search results. We set a maximum distance metric threshold for all found sequential matches, filter out all fingerprint matches from the same video, and decide on a sequence threshold (minimum duration in frames/seconds). Furthermore, we decide how many frames should be allowed to deviate from this threshold in a sequence (see allow\_skip configuration) and combine all overlapping sequences into one.

END HERMENEUTICS

### Step 6. Output matching results

Final matching results are shown in the form of frame thumbnails where the longest found sequences will be shown first. It is possible to limit the number of shown sequences and customize the frame size. A detailed example of what the matching results for the NASA videos may look like can be found in the article’s hermeneutics layer.

On the whole, the VRD’s matching results are meant to function as a guide that points users towards videos that might be interesting to study manually in more detail. In other words, we strongly advise against using the VRD’s matching results as an absolute proof of video reuse and instead emphasize the need to double-check the toolkit’s evaluations. In short, the VRD should be approached as an assistance tool in navigating large video datasets. As will become evident, this is not least due to the weaknesses and pitfalls that the software brings with it.

START HERMENEUTICS

We adjust the frame size and number of shown sequences and have a look at the final matching results. Note that the match mean distance values shown above the sequences may be higher than the previously set distance metric threshold. This is normal and expected, due to the merging of overlapping sequences.

END HERMENEUTICS

START HERMENEUTICS

NOTE TO PEER REVIEWER FROM THE AUTHORS: if you are reading this in the journal’s preview/rendering of the article you will be shown *a lot* of sequence matches below. We are aware that this creates a terrible reading experience and apologize for the inconvenience. At the moment, we are having a dialogue with the journal editors regarding the possibility to collapse cell outputs into a scrollable cell, which would make it much easier to read the hermeneutical layer. If you are reading this in Jupyter notebook, you will get a sense of what we want it to look like. If the journal is unable to meet our request to implement scrollable cell outputs, we are prepared to seriously revise how much data/output is shown in the hermeneutical layer. For now, however, we let the data remain as it is.

END HERMENEUTICS

In the matching results discussed below, we instructed the VRD to find similar-looking sequences between the NASA-videos that were at least 5 seconds long and had been given a distance metric value below 125. A maximum of 2 frames per sequence were allowed to deviate from this threshold. In the first iteration of matching results, the VRD found 12 sequences that matched these criteria.

The longest found sequence was 26 frames (or seconds) long and showed a clip where Neil Armstrong slowly climbs down a ladder and takes his first steps on the surface of the moon (figure 4). Vaguely discernable due to low resolution and a dark shadow cast by the Eagle shuttle, we see Armstrong wearing a white space suit in the lower left corner.

The second longest sequence depicts the Eagle shuttle’s descent toward the Moon, and includes panoramic scenes of the Moon surface (figure 5). The top left corner of the image is black since the camera’s view is partly blocked by the aircraft. Upon closer inspection, however, we find that the shown frames are not 100% identical. This illustrates the VRD’s (or rather, the Faiss library’s) particular ways of dealing with videos that contain slowly changing scenes. While the toolkit has, indeed, found frames with a high visual similarity, it is clear that we are seeing a matching result that appears to be a few seconds “off”. This is simply because the frames are similar enough to receive a comparatively low distance metric value. Theoretically, it would be possible to deal with this problem by increasing the distance metric threshold. However, doing so would likely imply that other interesting matches could get lost. In cases such as this, we therefore recommend accepting the somewhat imperfect matching result.

In the same sequence, we also notice that the VRD has matched several near-black frames (see the seven frames furthest to the right). This issue also occurs in other sequences, where near-black frames have been given a low distance metric value. While these matches are technically correct–indeed, the near-single-color frames do have a very high visual similarity–the matching results are practically irrelevant for studying video reuse. If matching single-color or near-single-color frames is a big problem in our dataset, we can deal with this problem by running an additional filter called ignore\_these\_frames (more about this soon).

When looking at the remaining matching results, we see that footage of the Eagle shuttle’s descent toward the Moon can also be found in three other sequences. This is due to the fact that the documentary The Moon and Us only reuses parts of the original footage. More precisely, we find that footage from the Eagle’s descent was shown for 20 seconds in the TV series episode The Moon and Us–starting 6 minutes and 23 secconds into the film. In the original video The Eagle Has Landed, however, footage from the Eagle’s descent is shown for roughly 2 minutes and 34 seconds (starting at 09 minutes and 3 seconds). Furthermore, very similar footage is shown for 1 minute and 8 seconds, starting 18 minutes and 20 seconds into The Eagle Has Landed. This time, however, it is not the Eagle’s descent, but ascent that is shown, as the shuttle begins its journey back to Earth (figure 6).

What has happened in our matching results, is that the same sequence of frames from The Moon and Us has been matched with different scenes depicting the Eagle’s descent and ascent from the moon in the The Eagle Has Landed. This issue is difficult to correct with any filter configurations since the footage does share many visual similarities. However, it can be helpful to know that when similar frames are found in different sequences, this can be an indication that similar footage appears several times within the same original video. Alternatively, it can indicate that footage has been cut/shortened in the process of reuse.

In the final sequence that includes footage of the Eagle’s descent (figure 7), the matching results are simply inaccurate. It is difficult to determine precisely why these matched frames have been given a comparatively low distance metric value when using CNNs, even without the complications added by using Faiss, and even attempting to do so is largely outside the scope of this article. One can speculate, however, that it has to do with the existence of vertical lines in the images.

The third longest sequence found in the matching results shows a series of still images that were taken during the collection of rock samples and the placement of scientific equipment on the moon (figure 8). Still photographs taken from the same scientific excursion can be found in three of the other sequences (figures 9,10 and 11). In the longest of these sequences, the matching results are accurate, although in the other sequences we once again see an example of how very similar footage has been mismatched.

Of the remaining sequences, the first is 8 seconds long and depicts Neil Armstrong’s and Buzz Aldrin’s view from the Eagle Shuttle (figure 12). The final sequence is 4 seconds long, and depicts footprints on the Moon surface (figure 13).

### Step 7. Run additional filters

After having studied the first set of found sequences, we can choose to apply some additional filters and rearrange the sorting of the final output. For instance, it is possible to remove unwanted sequences from the matching results, thereby ‘cleaning’ the dataset from uninteresting frames. This can for example be useful if long sequences of black (or near-black) frames are found in the search results. Likewise, it can be helpful if frames with text have been excessively matched, as this is a common problem.

It is also possible to change the sorting of the matching results, only show matching results for one or several selected videos, and re-adjust the distance metric threshold, shortest sequence threshold, and allowed skipped frames within each sequence.

START HERMENEUTICS

To remove unwanted sequences, we fill out an Excel template called “demo\_unwanted\_frames.xlsx” which is placed in the demo project folder, containing the columns “Video”, “Start time”, and “Duration”. We copy-paste the information to be entered in the columns directly into the Excel spreadsheet from the text shown above each sequence preview above. The “Video” column should contain information about the video name (including any extension such as .avi or .mpg). The “Start time” column should include information about the start time on a HH:MM:SS format (such as 01:55:22 for 1 hour, 55 minutes and 22 seconds into the video file), and the “Duration” column should state the duration of the sequence to ignore in seconds (e.g. 120 for 2 minutes). There is no problem with additional columns, as these will simply be ignored. When the Excel spreadsheet is finished and saved, we run the remove\_unwanted\_sequences filter.

END HERMENEUTICS

START HERMENEUTICS

When the filter has been applied, we can double-check what sequences were removed below. The filtered sequences will not appear again in later matching results.

END HERMENEUTICS

START HERMENEUTICS

As can be seen in the code cell above, two matched sequences were removed from the dataset.

END HERMENEUTICS

START HERMENEUTICS

Aside from removing unwanted sequences, it is possible to change the sorting of the matching results. Initially, matching results are sorted according to the length of sequences, but it is possible to adjust this to instead sort according to the lowest/shortest median distance metric value per sequence. We can also re-adjust the distance metric threshold, shortest sequence threshold, and allowed skipped frames within each sequence, or filter the search results so that only matches from one or several selected videos are shown. Prompts for applying additional filters can be found below.

END HERMENEUTICS

START HERMENEUTICS

To try them out we recommend you download the VRD and demo dataset on your local computer. For instance, if you want to show sequences that are a maximum of 5 seconds, sorted by shortest duration first, you run the code below. You can also decide to run more filters at the same time.

END HERMENEUTICS

### Possibilities and pitfalls

As shown in the demonstration with the NASA videos above, the VRD can function as an efficient tool for identifying instances of video reuse and the toolkit did help us find a series of reused sequences that are open for further explorations. For instance, how has footage from the moon landing in 1969 been reused in other documentary films and news broadcasts throughout history? When, where, and how has this footage played a role in extraterrestrial imaginaries, or potentially helped shape narratives concerning fundamentally different topics? What footage from the Apollo 11 mission has become most iconic (i.e. most excessively reused) throughout history? And which scenes from the space mission have been excluded from later documentary/news productions, thereby becoming partially erased from cultural memory? Naturally, the investigation of the Apollo 11 mission would have been more interesting, had the toolkit been applied to a larger dataset, or perhaps an entire archive. Thus, we suggest there is great potential for archives and cultural heritage institutions to consider fingerprinting their collections, thereby opening them up for new ways of search.

On a more practical level, the VRD toolkit works very well for finding visual similarities in video collections that have a high resolution and similar color/contrast scheme. The lesser original materials have been modified, edited, and/or remixed, the easier it also is for the toolkit to find relevant similarity matches. There are, however, a couple of instances where the toolkit performs less well. One such instance is when the VRD is applied to source materials that contain lots of textual and symbolic overlays, such as subtitles, news show banners, and/or tv-channel symbols. Another problem is videos that contain a lot of single-color (or near-single-color) frames. This is because text, symbols, and (near-)single-color frames tend to receive very low distance metric values. While this is often technically correct – since the pictures often do look very similar - it may cause problems in the search for cultural reuse.

For instance, finding out that several black frames share visual similarities is of little use-value in research situations. Likewise, it is comparatively uninteresting to find out that frames with text look alike, especially when the text in such images does not contain the same words or even language, which is often the case. Similar problems also arise with symbols that are mistaken for other symbols, even though they share few visual similarities. If a recurring problem in the matching results is images with texts, symbols, or (near-)single-color frames, we suggest using the function remove\_unwanted\_frames to clean the dataset (see hermeneutics layer). This can be applied by manually selecting precisely what sequences to remove/ignore in the matching results, or by applying a general rule for ignoring frames in all videos. For instance, if a recurring problem in the matching results is frames showing video introductions and after texts (sections that often contain text, symbols, and black frames), it might be worth considering removing a fixed number of frames from the beginning and end of each analyzed video. However, this will always come at the expense of losing possibly interesting matching results.

The VRD’s embeddedness in Jupyter Notebooks also brings with it some possibilities and pitfalls. While Jupyter Notebook provides excellent opportunities to document workflows and comment on code, the software’s memory storage carries some drawbacks. More precisely, Jupyter Notebook stores information about executed code during each session, so that it is possible to work in a non-linear way, jumping back and forth between code cells without necessarily re-running the entire notebook (*van Es, Schäfer, Wieringa 2021* p. 58). This can be very useful, but it also means that if code is executed and later deleted from a notebook during a session, the deleted code will still be saved in memory and continue to influence the analysis. For instance, this means that other researchers may have difficulties reproducing the results. To ensure that deleted code cannot influence the analysis, it is prudent to re-run the entire notebook regularly.

Aside from these practical issues, a couple of broader pitfalls and ethical issues regarding the use of CNNs must also be considered. We will return to a discussion concerning these issues in the [concluding remarks](#concluding-remarks). For now, however, let us have a look at an example of how the VRD can be used to study audiovisual reuse based on a larger dataset.

## Finding reuse in the SF archive

As mentioned in the introduction to this article, the SF-archive – in Swedish, *Svensk filmindustris journalfilmsarkiv* – is one of the most important audiovisual archives in Swedish film history. Consisting of film fragments, newsreels, and short films dating from 1897 to the early 1960s, the archive was digitized in the early 2000s and was previously used as a source in numerous TV programs and compilation films. In what follows, we re-trace the history of the SF-archive and reflect on how this archive has been reappropriated and remixed throughout more recent media history. More specifically, we explore *how* and *when* 1,400 of the oldest film clips that can be found in the archive appear in a series of historic TV documentaries that are available on RosaMannen’s YouTube channel. After being uploaded on YouTube, we are thus studying SF footage that has gone through double remediation, since the footage was first included and broadcasted in historic television documentaries, and then digitized and made openly available online.

The selected videos from the SF-archive vary in lenght but are usually between five to fifteen minutes long. Hence, our SF-dataset contains approximately 10,000 hours of moving image material. Regarding the TV series, we chose to work with 34 episodes taken from four different series, all focusing on Swedish 20th-century history. Put together, our TV-dataset consists of roughly 30 hours of film. When preparing the TV dataset, we edited out all introductory and closing texts from the films. This was done to minimize the risk of finding excessive matches with texts and symbols in the matching results ([see discussion in section Potentials and Pitfalls](#anchor-potentialls-and-pitfalls)). Most (but not all) of the TV-series episodes can be found on RosaMannen’s YouTube channel, and missing episodes were added through The Swedish Media Database ([SMDB](https://smdb.kb.se/smdb/english/)) which lets university staff order audiovisual content, giving access to streamed copies. All episodes in our selected TV series contain footage from the SF-archive – but also interviews, talking heads, stills, or other inserted materials. Hence, we knew that the VRD we would detect several visual similarities, but that there would also be sequences without any matches at all.

The first TV series, entitled [*The Time of Expectations*](https://www.youtube.com/watch?v=LMh5Jd9OS0I) (*Förväntningarnas tid*) — with an unknown producer — was broadcasted on Swedish Television during autumn 1985. The series consisted of four episodes that depicted the economic and political developments in Sweden following World War II and explored the landscape of opportunities that opened up when the country emerged as one of few European countries with an undamaged industry — and population — due to the country’s political neutrality. The series is foremost based on footage from the SF-archive but also contains a number of interviews, featuring well-known Swedish intellectuals and public figures such as novelist Per Olov Enquist and the political scientist Gunnar Myrdal.

The second TV series, *Gold Nuggets* (*Guldkorn*) — sometimes with the addition “from SF” or from a particular year as [1945](https://www.youtube.com/watch?v=xKV1B3POqn4) — was a long-running series broadcasted on public service television in Sweden from the mid-1980s up until the early 2000s. All episodes were produced by Jan Bergman who worked at the film archive at Swedish Television for many years. Bergman’s *Gold Nuggets* combined a deep knowledge of the SF-archive, often detecting and inserting unusual footage, with a more traditional style of filmmaking. Characteristic of his TV-series is a combination of his own personal narration with that of the typical fast-paced voice-over taken from the original newsreels in the SF-archive. In the dataset, six hour-long episodes of *Gold Nuggets* from 1985 and six from 2002 are included.

Bergman — together with Britta Emmer and Ebbe Schön — was also the producer of the third selected TV series, entitled [*Sweden, A Long Time Ago*](https://www.youtube.com/watch?v=MettI_h2Oiw) (*Sverige, för länge sedan*) broadcasted in 1984 in ten short episodes. The series — depicting different aspects of Sweden and its 20th-century history — is an illustrative example of how instrumental the SF-archive was during the 1980s. The series (in the form of individual films) were produced for the Swedish Educational Broadcasting Company (a subsidiary to the public broadcaster) and were clearly aimed as a kind of filler for school TV. “Compiled of material from the SF-archive”, was explicitly stated in all of the film’s closing credits. As in *Gold Nuggets* Bergman’s voice-of-god sutured the footage.

If the series *Sweden, A Long Time Ago* was all likely produced fairly rapidly — a budget compilation of sorts with its ten short episodes — the opposite is true of the final TV series that was selected for our experiment, namely the eight hour-long episodes of [*A Hundred Swedish Years*](https://www.youtube.com/watch?v=s53UPoAuh50) (*Hundra Svenska år*), first broadcast in 1999. The latter series was directed by Olle Häger, a long-time TV-producer with a career in public service television spanning three decades. Häger — and his narrator Hans Villius — are arguably the best-known documentary filmmakers in Sweden, having worked for decades with [still photography productions](https://www.youtube.com/watch?v=09QtLKchRq0) and films made from the SF-archive. *A Hundred Swedish Years* is usually regarded as one the finest historical documentaries compiled from the SF-archive. The production was a lavish one with a major budget and also included numerous interviews with elderly Swedes (recorded in the mid 1990s). The eight episodes each have a distinct theme — politics, fashion, labor market, children, technology, vacation, monarchy and foreign policy — giving the series a changing and dynamic setting. To prepare his TV production, Häger worked for years with the SF-archive, and even reprinted some footage with a higher resolution. *A Hundred Swedish Years* was broadcasted on prime time during the autumn of 1999, and was later also commercially released on both VHS and DVD.

In our exploration of the reuse taking place between the SF-archive and these four TV series, we perform two different similarity searches. First, we compare all selected television series episodes against each other to see if they contained any internal examples of reuse. Does any particular footage appear to have been reused multiple times in the documentaries? For instance, can we find examples of Jan Bergman and Olle Häger reusing the same footage? Second – and arguably more interesting – we compare the 30 hours of series episodes against the SF-archive (ignoring any potential examples of reuse that might be discovered *within* the TV series and the SF-archive respectively). In short, we wish to retrace the steps of the documentary filmmakers and map their choice of selected historic footage. Since our dataset from the SF-archive contains some 10,000 hours we are naturally interested in exploring what type of footage that was reused the most. Was it primarily the oldest footage? Or were other selection and editing criteria (aesthetic, historical, topical etcetera) of principal interest?

### Reuse in the televisual dataset

Producers of TV programs usually work within a limited time frame and the same goes for documentary productions. Hence, one might assume that filmmakers do their research primarily by looking at previous (similar) productions to save time. However, if the topic is general – i.e. Swedish history – there is plenty of footage to choose from, and reusing sequences that have already appeared in previous documentaries might come across as awkward. Then again, it is of course hard (if not impossible) for television audiences to remember if a sequence is shown for the first time or not. After applying a series of filters and features to customize the matching results, we were not surprised to find several examples of reuse within the televisual dataset.

START HERMENEUTICS

We start a new project called “tv\_series”, extract one frame per second from the TV dataset, and apply the neural network EfficientNetB4 in our analysis.

END HERMENEUTICS

START HERMENEUTICS

We then apply the selected neural net to compress and vectorize the frames into fingerprints and use Faiss to index and then calculate the most similar neighbors for each fingerprint.

END HERMENEUTICS

START HERMENEUTICS

To get a sense of where to apply a distance metric threshold, we have a look at the initial distance metric distribution in a histogram.

END HERMENEUTICS

START HERMENEUTICS

Based on this histogram, we decide to begin by applying a distance metric threshold of 120 and decide to filter out all matching frames that have origins in the same video. We also decide to proceed by only searching for sequential matches that are at least five seconds long with an allowed skip rate of two frames per sequence. We merge overlapping sequences into one. As can be seen in the statistics below the code cell, this reduces the number of found sequences from 14 571 to 1 302.

END HERMENEUTICS

START HERMENEUTICS

We also decide that we are not interested in analyzing sequences that come from the same TV series. For instance, one TV series production team may have reused the same video segment several times in different episodes, but such examples of reuse are uninteresting to us since we want to explore potential instances of reuse among *different* TV series producers. To do this, we apply a customized filter called not\_from\_same\_tv\_series. This filter will be applied in all proceeding displays of matching results.

END HERMENEUTICS

START HERMENEUTICS

Next, we look at the remaining matching results in a stacked bar plot, where the reuse count on the Y-axis refers to the number of found sequences that are at least 5 seconds long.

END HERMENEUTICS

START HERMENEUTICS

As can be seen in the bar plot, it appears as if footage from primarily two videos has been excessively reused within the TV dataset. This concerns the episode *Vägen genom krisen* (The path through the crisis) from the TV series *Förväntningarnas tid* (The time of expectations) and the episode *Nu har jag kastat min blå overall* (Now I have discarded by blue overall) from the TV series *Hundra Svenska år* (A hundred Swedish years).

END HERMENEUTICS

START HERMENEUTICS

We proceed by looking at the matching results as thumbnail previews. The quality of these matches will give hints regarding how well the distance metric threshold was set in the SequenceFinder step above. If we find that many sequences that do not match, perhaps the distance metric threshold is set too high. Similarly, if all matches look good but a bit short, perhaps the distance metric threshold is set too low. In other words, we can use this data to adjust the previous filtering settings according to our specific dataset. As previously noted, the distribution of distance metric values will for example be impacted by the choice of neural net, the layer from which to extract the fingerprints, the size of the dataset, and the quality of the source material. We recommend finding a threshold that is slightly too high (rather than too low), to avoid missing relevant matches. Invalid matches can then be filtered out using other tools. In this notebook, we limit the number of shown sequences to 10 in the example below. During our analysis, however, the 200 longest sequences were analyzed.

END HERMENEUTICS

START HERMENEUTICS

When studying these thumbnails, we decide to keep working with a distance metric threshold of 120 but notice that a special type of frame seem to distort the matching results. These are frames showing talking heads – that is, shots of people’s heads and shoulders as they are talking to the camera. As can be seen in several examples, such frames have been given a comparatively low distance metric, even though they depict entirely different people. We decide to filter out these sequences manually, even though this means that actual reuse of talking heads may get lost. This is done by adding the unwanted sequences to an Excel (.xlsx) template entitled “unwanted\_frames” and running the remove\_unwanted\_sequences function. We also find some black (or near-black) sequences and sequences containing text in the matching results and add these to the list of unwanted frames as well. As can be seen in the statistics after the code cell below, this reduces the number of found sequences from 402 to 105.

END HERMENEUTICS

START HERMENEUTICS

To double-check what sequences were removed in the previous step, we run the code below. This will provide a sample 50 sequences that were filtered out using the unwanted\_sequences feature.

END HERMENEUTICS

START HERMENEUTICS

As can be seen in the thumbnail preview above, the Excel spreadsheet did include many talking heads, as well as black or near-black frames. Interestingly, we also find a couple of correct matches. For instance, the first sequence does appear to be correct, even though it has a slight time shift.

END HERMENEUTICS

START HERMENEUTICS

Next, we look at our updated matching results in the form of a stacked bar plot.

END HERMENEUTICS

START HERMENEUTICS

As can be seen in the bar plot, the number of found sequences is now highly reduced. Still, however, we find the same two TV series episodes being seemingly reused the most amount of times (i.e., the episodes *Vägen genom krisen* and *Nu har jag kastat min blå overall*).

END HERMENEUTICS

START HERMENEUTICS

We have a look at the remaining matching results as thumbnail previews once more. The longest found sequence will be shown first, and we decide to only show the 200 longest sequences found.

END HERMENEUTICS

START HERMENEUTICS

When looking at these thumbnails, we notice that a few unwanted sequences remain in the matching results. If desired, we could have gone back to add these to the unwanted\_frames Excel spreadsheet and filtered the matching results once more. For instance, one sequence shows mismatched newspaper pages, and in another, clouds appear to have been mixed up with trees. In this case, however, we decide that the dataset is sufficiently clean to facilitate the rest of our analysis.

END HERMENEUTICS

START HERMENEUTICS

Next, we decide to drill deeper into the matching results by having a closer look at reuse based on some specific TV series episodes. Using the latest bar plot, we begin by looking at the episode with the most identified instances of reuse (i.e. the video entitled “Förväntningarnas tid\_Vägen genom krisen”).

END HERMENEUTICS

START HERMENEUTICS

We also have a closer look at all matching results for the four clips that can be found furthest to the left in the bar plot above.

END HERMENEUTICS

In total, 89 169 fingerprints were produced from the TV dataset and compared against each other. Among these fingerprints, we applied a distance metric threshold of 120 (see [step 4](#X693739fce065c61b8a420990b6fbf89abcb32a1)) and filtered out all matching frames with origins in the same TV series. We also searched for sequential matches that were at least 5 seconds long and allowed a skip rate of two frames per sequence (see [step 5](#step-5-filter-matching-results)). We also filtered out 297 uninteresting sequences that were identified manually.

In the remaining matching results, we for example found that several cooking sequences with children had been reused in both *Gold Nuggets* from 1946, and the episode ‘The Path Through the Crisis’ (Vägen genom krisen) from the series *The Time of Expectations* (Förväntningarnas tid). Looking closer at the footage (see figure 14-17), it was apparent that the children doing the cooking were boys. The Swedish welfare state did not solely deem domestic sciences to be a subject for girls – gender neutrality was to rule the kitchen – which naturally was an alluring theme for TV-producers. The selected sequences were almost identical, and the boys – of course, supervised by a female teacher – were preparing pancakes and apple pie: “nicer to eat than cook”.

Upon closer inspection, we also found that footage from the same episode, ‘The Path Through the Crisis’, could be found in no less than five episodes from the TV series *Gold Nuggets*. More precisely, footage from ‘The Path Through the Crisis’ re-appeared in *Gold Nuggets* depicting the years 1941, 1945, 1946, 1947 as well as 1948. In these sequences, we found footage of refugees during the Second World War (figure 18), footage from the Swedish parliamentary election in 1948 (figure 19), and clips showing a newly-wed couple’s unpacking of a furniture series by the designer and architect Elias Svedberg, which was designed to be assembled in the home (figure 20). Since the producer of *Gold Nuggets*, Jan Bergman, worked at the Swedish Television film archive, he was well aware of the series *The Time of Expectations*. He might even have assisted in the production by suggesting interesting footage from the SF-archive. Hence, even if the latter series was screened on television the same year as *Gold Nuggets* (in 1985), Bergman did apparently not hesitate to include identical sequences in his own TV programs.

In a similar way, and following the plot below, we find indications that footage found in episode six of *A Hundred Swedish Years*, entitled ‘Now I have discarded my blue overall’ (*Hundra svenska år*, ‘Nu har jag kastat min blå overall’), has been matched nearly 25 times with content in other TV series episodes. Since Olle Häger produced his documentary series *A Hundred Swedish Years* in the 1990s, he had all likely seen most of the previous TV series in our experiment, cherry-picking the best footage.

Importantly, however, the plot above should not be read as an absolute proof of reuse, since it contains several mismatched sequences. These inaccurate sequences remain in our matching results since we intentionally decide not to apply harsher filtering settings, as this could risk filtering out correct matches (for further discussions on this topic, see the hermeneutics layer). In other words, the chart should be read as a tool for getting an overview of *where to start looking for* actual instances of reuse. Following this chart, we find an eight second long identical match between Häger’s production and the transportation episode in *Sweden, A Long Time Ago*, featuring a sequence from 1913 with automobiles and trams in the city of Stockholm. In the background of the footage a horse-drawn hearse suddenly appears – *fin de siècle* Stockholm was still a city with tens of thousands of horses (figure 21). Häger’s voice-over comments on the hearse, steering the gaze towards the background of the image.

Another match from his episode displays a similar pattern, namely a short sequence of some five seconds depicting the cinema Pariserteatern (on Roslagsgatan in Stockholm), footage which also appears in two episodes from Jan Bergman’s *Sweden, A Long Time Ago* (figure 22). In one of Bergman’s episodes, the focus is on education and children with Sweden’s elementary school system being described as harsh: “so it was not surprising that children were dreamy, escaping from everyday life by going to the cinema”, the voice-over states. In another of Bergman’s episodes, focus is placed on leisure activities and the footage is used to exemplify how local cinema’s provided a new source of entertainment and rapidly spread across the country. In contrast, Häger uses the identical sequence to suggest that the film operator at the cinema Pariserteatern (visible in a window above the entrance) only had a few minutes for himself, again guiding the viewer to a significant detail within the footage. The point to be made is that even if these sequences are identical, by using a voice-over to accentuate different aspects within the footage, it is possible to alter its meaning significantly. Discovering how footage is re-contextualized and embedded in alternative meanings throughout time, is thus a good example of where the VRD can be of assistance.

Then again, the VRD also indicates a false match with regards to the sequence with the film operator at Pariserteatern (figure 26). It is claimed that it resembles a six-second sequence of a factory in the series *Time of Expectations*. At first glance, the sequences do look familiar with people in the foreground of the image and a building with a diagonal shape in the center. But the sequences are in fact not identical at all. Hence, a word of caution should be raised regarding the plotabove. When looking at the plot, it appears as if sequences from *A Hundred Swedish Years*, ‘Now I have discarded my blue overall’ (*Hundra svenska år,* ‘Nu har jag kastat min blå overall’) have been reused nearly 25 times. Upon closer inspection, however, several of these sequence matches are of doubtful quality. More precisely, 17 of the matched sequences are inaccurate and display things such as mismatched people, near-black frames, and frames depicting clouds or trees. This illustrates that plots such as the one shown in figure 21 may say a lot about the *quantity* of matches, but little about their *quality*.

For instance, the VRD – and its underlying CNNs and ways of finding similarity neighbors using Faiss – sometimes suggests reuse regarding sequences that are of an abstract nature and depict environments, buildnings, streets etc. Such footage shares graphical similarities (like the cinema and the factory in the example above). Sometimes these matches are correct, as the almost symbolic curved road in a longer dolly shot, filmed from a car depicting sheep running along a village road in rural Sweden around 1910 (figure 24). This sequence appears in both Häger’s *A Hundred Swedish Years* and Bergman’s *Sweden, A Long Time Ago*. But at other times, comparable abstract scenes are totally dissimilar, even if the VRD defines the frames as looking alike: trees, clouds, or shots of the sky are three examples (figure 25), inserted sequences of books or newspapers another (figure 26). Close-ups of newspaper pages in *Gold Nuggets* and *Time of Expectations* are hence perceived as having a close visual similarity by the software – even if it is apparent to the human eye that these newspapers have nothing in common at all, other than being newspapers.

To summarize, we can thus state that aside from a series of mismatched frames, the VRD has indeed helped us identify multiple instances of reuse within the televisual dataset, of which we have only highlighted and discussed a few in this narrative layer (for a closer look at the matching results, see hermeneutical layer). One conclusion to be drawn from the findings is that the producers of the selected TV series showcase a somewhat surprising tendency to reuse the same audiovisual content, despite having access to thousands of archival films at the television archive. It is apparent that some sequences have been particularly enticing for producers, especially those that depict the film medium *per se* (for example the cinema Pariserteatern). The same goes for striking footage of everyday life in the Swedish capital; nonfiction film sometimes displays a salient aesthetic beauty. As we have found the same audiovisual content reemerge in different TV series, we have also been able to start mapping how footage is assigned different meanings – particularly via sound. Early cinema was mute, but television documentaries nearly always added artificial sound, a distortion of sorts but also a way of altering the significance of the film footage. The VRD does not detect sound, but as we have shown it can be used to highlight how sound changes filmic intentions at scale. Undoubtedly, the task of manually watching and comparing each frame in these 35 episodes (roughly 30 hours of television) would have been insurmountable for any single human being, let alone a modestly sized research team. This will only become more apparent as we compare the televisual dataset with the 1,400 selected newsreels and short films from the SF-archive, a step that will increase the number of potentially matching frames exponentially.

### Reuse in the TV and SF dataset

Next, we want to find visual similarities between the televisual dataset and the SF dataset, without searching for examples of reuse within the SF dataset as such. Put simply, we are not interested in studying internal reuse in the SF archive, since all likely such reuse was modest, occurring primarily when Swedish Film Industry produced documentaries or short films of past notable events – as when SF in 1956 celebrated one of its cinematographers, [Gustaf Boge](https://www.filmarkivet.se/movies/gustaf-boges-samling-nr-1/), having worked for the company during 35 years. We hence decided to focus our efforts on exploring which footage from the SF-archive was reused in the aforementioned television series. As stated, the SF dataset is considerably larger than the televisual dataset, and to perform this kind of similarity search we first had to create two sets of indexed fingerprints. Secondly, we combined the datasets and used Faiss to calculate the closest similarity neighbors between the two. Since the TV dataset had already been fingerprinted and indexed, we were able to reuse previous data, but we still needed to process the videos from the SF dataset.

START HERMENEUTICS

We begin by starting a second project called “sf\_archive” and extract one frame per second from the SF dataset. Again, we apply the neural network EfficientNetB4 in our analysis.

END HERMENEUTICS

START HERMENEUTICS

Next, we compress and vectorize the extracted frames into fingerprints using the selected CNN.

END HERMENEUTICS

START HERMENEUTICS

We index the fingerprints using Faiss.

END HERMENEUTICS

START HERMENEUTICS

And then combine the two projects using the combine\_projects feature. This will initialize the use of Faiss to calculate the most similar neighbours between the indexed fingerprints in SF dataset and TV dataset.

END HERMENEUTICS

START HERMENEUTICS

When the similarity neighbors have been calculated, we look at a histogram of the distances metric value distribution and use it to determine a reasonable threshold for finding sequences.

END HERMENEUTICS

START HERMENEUTICS

Based on the data shown in the histogram above, we decide to apply the distance metric threshold 120 and filter out all matching results from the same video. We also decide to only search for sequences that are at least five seconds long, and allow a skip rate of 2 frames per sequence. As can be seen in the statistics below, this reduces our dataset from 10 670 to 1983 found sequences.

END HERMENEUTICS

START HERMENEUTICS

Using the same rationale as before, we filter out the same sequences that were labeled as unwanted in the TV dataset. Since we will only match the TV dataset against the SF dataset - and not search for any examples of reuse within the SF dataset as such - the removal of these frames should be enough to clean the search results. Note, however, that if any real reuse of unwanted frames has occurred (for instance, reuse of footage containing talking heads), these sequences will be lost from the matching results. As can be seen in the statistics below the code cell, this reduces the number of found sequences from 1983 to 1553.

END HERMENEUTICS

START HERMENEUTICS

To double-check what sequences were removed in the previous step, we output a sample of 50 sequences that were removed using remove\_unwanted\_sequences functionality.

END HERMENEUTICS

START HERMENEUTICS

Looking at these thumbnails, we notice that at least one accurate match has been removed. This concerns the first sequence shown above. Other than that, however, the removed sequences do seem to contain a large number of wrongly matched talking heads, or uninteresting frames.

END HERMENEUTICS

START HERMENEUTICS

We look at the remaining matching results in a stacked bar plot.

END HERMENEUTICS

START HERMENEUTICS

As can be seen in this plot, footage from the video SF27891.1 appears to have been reused the most amount of times, as over 35 sequences from the video have been identified in 11 of the TV series episodes. More than 15 matching sequences have also been found in the videos entitled SF2410A-C.1, 2764.1, 2554.1, 2183.1, and 2831.A-C.1. We take note of these video clips and will return to them later.

END HERMENEUTICS

START HERMENEUTICS

We also output the matching results as thumbnail previews, sorted with the longest found sequence first. We limit the number of shown matching results to the 200 longest sequences. To enlarge the thumbnail previews when the notebook is viewed in Jupyter Notebooks, it is possible to double-click on the sequences. In the Journal of Digital History’s rendering of the text, it is possible to zoom in on the thumbnails by opening them in a separate window by right-clicking on the images, or copying/pasting the link into a new browser window, for example.

END HERMENEUTICS

START HERMENEUTICS

When browsing through the thumbnail previews, we do find that most of the sequences look correct. In some cases, there is a slight time shift in the identified frames, but overall, it seems as if the VRD has successfully found several instances of reuse. Some notable exceptions, however, include sequences where crowds of people have been mismatched. We also see examples of talking heads that remain in the search results. Other mismatched sequences include frames showing ocean horizons, sailing boats, and snowy landscapes. We also find examples of frames that have likely been mismatched since they are dark or near-black.

END HERMENEUTICS

START HERMENEUTICS

We add these and other found unwanted sequences to a new Excel spreadsheet called “Unwanted\_frames\_2” and save it for later. Notably, adding unwanted sequences to the document will imply that our abilities to identify *actual* instances of reuse of footage showing talking heads, ocean horizons, trees and near-black frames, etc. will be severely reduced from now on. This is a price that we decide that we are willing to pay, however, to ‘clean’ and make it easier to read and navigate in our matching results.

END HERMENEUTICS

START HERMENEUTICS

An alternative to using the feature for removing unwanted frames could have involved lowering the distance metric threshold. This, however, could also imply that correct matches are removed from the search results, for example if a “correct” sequence match may be given a lower distance metric value than an incorrect sequence. This can happen if the color or resolution is very different in two video clips that otherwise contain the same footage. In this experiment, we choose to work with the feature for removing unwanted frames since it gives us a clearer idea of which matching results may have been removed from our dataset, such as sequences showing talking heads and persons depicted in half-figure shots.

END HERMENEUTICS

START HERMENEUTICS

To further inspect the search results, we look at the 200 longest sequences from the 5 most reused videos, as seen in the stacked bar plot above. We begin by looking at the matching results for the video SF2789.1, that is, the video clip with the most identified instances of reused sequences.

END HERMENEUTICS

START HERMENEUTICS

In this case, we find that a particular sequence from the video SF2789.1.mpg has been excessively mismatched in the search results. As it turns out, this sequence shows Prince Wilhelm of Norway and Sweden reading a poem called “While the boat is drifting” (“Medan båten glider”).

END HERMENEUTICS

START HERMENEUTICS

We add the sequences showing Prince Wilhelm and other talking heads to the Excel spreadsheet “TV\_unwanted\_frames\_2.xlsx” and continue by studying the second most reused video clip in the same way, zooming in on the matching results for the video SF2410A-C.1, which depics a [military exercise](https://smdb.kb.se/catalog/search?q=SF2410&x=0&y=0), performed by the Swedish Navy in 1918.

END HERMENEUTICS

START HERMENEUTICS

In this case, we find that frames depicting an ocean horizon have been excessively matched against ocean horizons in other TV series episodes. While this is technically correct – since the ocean horizons do look very much alike – it is uninteresting for our purposes here. However, skimming through the matching results we do find two interesting sequences. The first depicts an explosion on the ocean horizon, although with a slight time shift. The second sequence shows a correct match between shots of land, taken from a boat.

END HERMENEUTICS

START HERMENEUTICS

We add the remaining uninteresting ocean horizon sequences to the “Unwanted\_frames\_2” spreadsheet. We also look through the matching results for the video SF2764.1, SF2554.1, and SF2831A-C.1 and add unwanted frames found there to the same Excel document. In the interest of space, however, these matching results are not displayed in this notebook.

END HERMENEUTICS

START HERMENEUTICS

When the updated Excel sheet is finished, we apply the remove\_unwanted\_sequences function once more and output a sample of what the removed sequences look like to double-check the filtering.

END HERMENEUTICS

START HERMENEUTICS

We look at the remaining matching results in the form of a bar plot once more.

END HERMENEUTICS

START HERMENEUTICS

And output the remaining matching results as thumbnail previews. Once more, the longest found sequence will be shown first and we limit the number of shown sequences to 200.

END HERMENEUTICS

START HERMENEUTICS

Last, we also have a look at thumbnail previews in the 5 remaining SF clips that have been most frequently reused, showing the most reused clip first.

END HERMENEUTICS

After applying the same filters and features to customize the matching results (see hermeneutics layer for methodological details) we, unsurprisingly, found several reused sequences between the SF dataset and the television series. In our dataset, the titles of films in the SF-archive are always numerical (SF2001 etcetera and sometimes with letters SF2410A–C indicating sequels), yet the titles can be manually decoded and identified with the help of The Swedish Media Database ([SMDB](https://smdb.kb.se/smdb/english/)).

A first thing to note – looking at the plot above – is that sequences from the film SF2789.1 appears to have been reused the most. Over 35 sequences from this film were identified in 11 of the TV series episodes. SF2789 is, in fact, a short film from 1930, [*Medan båten glider*](https://www.filmarkivet.se/movies/medan-baten-glider/) (*While the boat is drifting*) featuring Prince Wilhelm, a royal Swedish media celebrity, himself an avid filmmaker (and author). In the film, Prince Wilhelm recites his poem “While the Boat is Drifting”, evocing northern Scandinavia’s “sparkling mountain lakes and mile-wide desert forests with their strange and enchanting mysticism”. However, unfortunately for Prince Wilhelm, his recital was not actually reused since the VRD excessively mismatched the medium shot of Wilhelm (from his knees up). For instance, the VRD inaccurately suggested that he was both identical to an unknown woman in Häger’s *A Hundred Swedish Years*, as well as (somewhat surprisingly) social democratic finance minister Ernst Wigforss (in the same TV series), and particularly so in a medium shot of him taken from a famous scene in Swedish film history with Wigforss debating God in a [1946 election film](https://www.filmarkivet.se/movies/per-albin-svarar-valkomna-till-oss/).

After a second round of processing our matching results and manually removing a series of mismatched frames, such as the one showing Prince Wilhelm above, we are left with a slightly more accurate graph of the final matching results (see figure 32 below). In this process, we also removed a series of manually identified mismatched sequences containing (near-)single-color frames, and frames showing easily misinterpreted scenes such as ocean horizons, trees, and clouds. Again, however, the plot should not be read as an absolute proof of reuse, since it still contains mismatched sequences that are kept to avoid filtering out correct matches. In other words, we once again approach the plot as a tool for getting an overview of *where to start looking for* actual instances of reuse.

Since all of our historical documentaries were based on the SF-archive – to a greater or lesser degree – the reuse depicted in the plot above did not come as a particular surprise. But which type of footage was reused, and what kind of sequences were favored by the filmmakers? The VRD helps us untangle some of these questions. Following the chart above, for instance, the VRD suggests that the film SF2836A – *Gustaf V in memoriam*, a short film about the Swedish king Gustaf V from 1950 – was reused in ten different documentary episodes within our televisual dataset. A number of these matches are incorrect but some, such as *Gold Nuggets* (from 1946), reused a 13 seconds long sequence from the SF archive, and so did the TV series *Time of Expectations*.

The reason for the popularity of SF2836A was likely twofold: on the one hand, it focused on the royal family – a frequent topic of newsreels during the first half of the 20th-century – and on the other hand, SF2836A was described as “a memory film” (compiled by filmmaker Nils Jerring) of the life and times of Gustaf V who passed away in 1950. In short, SF2836 gave later filmmakers the opportunity to find earlier footage of a popular subject, as well as more contemporary sequences. In Olle Häger’s episode of the royal family in Sweden, [“Å dyre prins”](https://www.youtube.com/watch?v=xYwA4EK8rog) (‘O, Esteemed Prince’, in *A Hundred Swedish Years*) at least one sequence was taken from SF2836A depicting a royal marriage (and its difficulties).

Another film from the SF-archive that features prominently in the reused footage is SF2410A-C. The VRD claims that sequences are reused eighteen times in seven different episodes in our televisual dataset. There are many mismatches, however. SF2410A-C depicts scenes from the Swedish navy in 1918, and Häger reused a sequence of explosions at sea in his last episode of *A Hundred Swedish Years*. Interestingly, the VRD suggests reuse of SF2410A-C within Bergman’s [*Gold Nuggets* (from the year 1941)](https://www.youtube.com/watch?v=x5W2jTeFYfM), thus indicating ahistorical representation. Of course, documentary films are not always historically accurate, especially concerning added sound or the usage of *iconic* footage (to use C.S. Peirce terminology). The VRD claims (amusingly) that scenes of the Swedish navy in 1918 are similar to the Soviet fleet during the Second World War, which is obviously not correct.

Accurate matches are, however, found for SF2221A-J, a numerical title that refers to Gardar Sahlberg’s compilation film *When the Century was Young* (När seklet var ungt) from 1961. This is not surprising since Sahlberg’s film was based on old newsreels and film fragments from the SF-archive, and espcially the oldest footage from before the First World War. In fact, Sahlberg is the person who during the 1960s was responsible for cataloging and preserving the SF-archive once it was bought by Radio Sweden from SF in 1954. As an homage to Sahlberg (one might assume) both Bergman in Sweden, *A Long Time Ago* and Häger reused SF2221A-J a lot. What we discover with the help of the VRD is thus a kind of meta-reuse or media archaeology of film historical layers, with TV documenataries reusing footage from a historical compilation film based on the SF-archive. We are also reminded of how this footage is remediated once more, as RosaMannen has uploaded *Sweden, A Long Time Ago*, ‘Commerce’ on [YouTube](https://www.youtube.com/watch?v=ygZkTkTPaYs). In this context, the episode has triggered a discussion around the age of the footage: “Must be before 1914 when open market trade with meat was banned in 1914”, suggests the user @jonasglanshed.

One particularly interesting reuse concerns the film fragment SF2183. It contains footage from Stockholm shot in 1913 and 1914, with different film fragments depicting everyday life in an urban setting as well as sequences shot from two early ballon ascents over the capital. In his documentary *Sweden, A Long Time Ago* Bergman reused an almost thirty seconds long sequence from these aerial balloon shots. Häger also reused similar sequences in his episode [“I Have Seen Everything That Is New”](https://www.youtube.com/watch?v=cVtFOZMHZjA) of *A Hundred Swedish Years*, both the acent of the ballon and aerial shots of Stockholm from above.

It is hardly surprising that prolonged sequences shot from a balloon appears in these documentaries. The footage is simply stunning. In fact, SF2183 seems to have been a favorite archival film for both Bergman and Häger. The reuse that was previously described – featuring automobiles, trams, the horse-drawn hearse, and the cinema Pariserteatern – all have their origins in SF2183. Not only is the footage from SF2183 frequently reused, but the film also has an interesting archival history in itself. Following catalog information and metadata found at the National Library of Sweden, this thirteen-minute film includes five different types of footage: everyday life in Vasastan (a neighborhood in Stockholm), shots of the cinema Pariserteatern, various images from Stockholm, and balloon ascents in 1914 and 1913. Since there are no copyright restrictions, the Stockholm City Museum have included the separate films from SF2183 on their portal [Stockholmskällan](https://stockholmskallan.stockholm.se/sok/?q=SF%202183). Online it is hence possible both to watch the [sequences from Vasastan](https://stockholmskallan.stockholm.se/post/29570) in accurate frame rate (which is not always the case on YouTube) or the [balloon ascent from Stockholm’s Olympic stadium](https://stockholmskallan.stockholm.se/post/29565) in 1913. In the last film, however, it is apparent from the intertitle that it is not of Swedish origin. On the contrary, it is clearly a Pathé film production. The SF-archive, in fact, contains a lot of international films and film fragments which SF and later Radio Sweden claimed to have copyrights to – which they obviously did not.

## Concluding remarks

When Radio Sweden purchased the SF-archive in 1964, the sole purpose was reuse. Old films were to be compiled into new TV programs and given an afterlife in a novel medium, albeit in a re-edited form. Today, a similar reuse of moving images also occurs on digital platforms, as old television programs and films have found a haven on YouTube (and similar sites). RosaMannen’s ways of making Swedish TV history available are appreciated by many and have been tolerated by SVT, perhaps because his VHS copies are of lesser quality. Then again, YouTube is today also the platform for algorithmic upscaling experiments of predominantly older films (to avoid IPR issues). Material from the SF-archive has, for example, been reused and uploaded by [ColorByCarl](https://www.youtube.com/@colorbycarl), with footage remastered and upscaled to 4K and 60 frames per second using various AI and software toolkits. Even if film archivists usually detest machine learning algorithms, such upscaling has received immense attention: Denis Shiryaev upscaled version of footage from the SF-archive – shot by [Swedish Biograph in New York in 1911](https://www.youtube.com/watch?v=hZ1OgQL9_Cw&t=1s) – has been viewed more than twenty million times (all likely including a number of bots). Naturally, algorithms alter these films but so did prior TV-producers. A wider digital context of the Video Reuse Detector hence also includes computational reuse in other ways. Probably, an increasing amount of films from the SF-archive will surmount on YouTube – or as Abigail de Kosnik has stated: “Online, each media commodity becomes, at the instant of its release, an archive to be plundered, an original to be memorized, copied, and manipulated – a starting point or springboard for receivers’ creativity, rather than an end unto itself” (*De Kosnik 2016*).

In the example presented in this article, we included roughly 1,400 videos from the SF-archive. Taking the research further, it would of course be interesting to include a larger part of the archive, if not the entire archive which is divided into subsections; SF2000, SF3000 etcetera for all 5,500 videos. When we started working with the whole SF-archive, however, we found that too many computational resources were needed and had to scale down to only use the 1,400 videos within the first SF2000 category, containing the oldest film titles. SF2001A-B, for example, includes the very first Swedish film materials, shot by a local Lumière cinematographer at the Stockholm exhibition in 1897. These sequences are popular and have habitually been used in historical television documentaries since the moving images are the oldest depicting Sweden. To our surprise, however, they did not feature prominently in our televisual dataset. As our displayed results indicate, by using the VRD it hence is possible to discern which footage from the SF-archive that was reused – as well as missing, popular film sequences. In theory, it would even be possible to detect exactly what sequences from the SF-archive were most popular and reused by TV producers at Swedish Television (SVT) from the mid-1960s until today, should the datasets be enlarged.

On a practical level, the VRD toolkit does not fully automate the search for audiovisual reuse yet it evidently makes it possible to search for - and find - audiovisual reuse in datasets whose size far outreaches what any individual human being can process. While manual labor is still needed to interpret the match results (involving, for instance, double-checking filtering settings, sorting out unwanted frames, and iteratively tweaking and improving the similarity search), we hope to have shown that the toolkit can be of help when sorting through large archives of film. Since the VRD’s ways of searching for visual similarities is based on still images, it is worth noting that the basic features of the toolkit (using the CNN + FAISS library, as well as some of the code for filtering matching results) could also be applied to study the reuse of images in general (photographs, drawings etc.,).

There are, however, a couple of things to keep in mind when using the toolkit. For instance, the matching results presented in this article would likely change if another CNN was used, or if the fingerprints would have been extracted from another layer in EfficientNetB4. Similarly, using another indexation library than Faiss to calculate similarity neighbors – such as [Annoy](https://github.com/spotify/annoy), [Milvus](https://milvus.io/), or [Hnswlib](https://github.com/nmslib/hnswlib) – would likely impact the matching results. And at the moment, we have not done any performance tests and comparisons with these tools. All of this is to say that every application of the VRD will generate a specific set of outputs that are specific to the datasets and settings at hand.

Aside from these practical issues, a couple of broader pitfalls and ethical issues regarding the use of CNNs must also be considered. For instance, CNNs have been widely critiqued for being biased – primarily because of problems surrounding their training datsets. Most CNNs (including all neural nets that come with Keras and are embedded in the VRD) are trained and developed using the so-called ImageNet challenge. ImageNet is a database consisting of over 14 million images collected from the Internet, which have been classified by human beings according to a taxonomy called WordNet that was developed by scholars at Princeton University in the 1980s (*Denton, Hanna, Amironesei, Smart, Nicole 2021*). This database is used to teach CNNs how to analyze visual content, as well as backtrack and fact-check the model’s interpretations. Yet previous research has found that several sociodemographic groups – such as women, people of color, LGBTQ+ communities, and people with disabilities – are underrepresented in the dataset (ibid.). As a result, CNNs trained on ImageNet have been shown to carry racial, sexual, and heteronormative biases, as well as biases relating to functionality. Similarly, images from non-Western countries are heavily underrepresented in the ImageNet database, meaning that CNNs trained on ImageNet have difficulties recognizing non-Western environments (*DeVries, Misra, Wang, van der Maaten 2019*). For instance, this includes having problems recognizing basic non-Western household environments such as kitchens/bathrooms and the everyday objects, furniture, and utensils found therein (ibid.). Others have found that CNNs tend to perform “a capitalist and product-focused reading of the world”, as the ImageNet dataset and WordNet taxonomy is biased towards commerce and contemporary consumer products (*Pereira, Moreschi 2021*).

In addition to dataset biases, convolutional neural networks also carry a series of “perceptual biases” that concern their fundamental ways of representing the world (*Offert, Bell 2021*). For instance, CNNs are trained to make classifications, and classifications by necessity involve reducing and limiting interpretations of what exists (*Bowker, Leigh-Star 1999*). As Kate Crawford and Trevor Paglen argue, each layer in a CNN is “infused with politics” in the sense that it is programmed to assume that fixed, universal and consistent concepts exist including, for example, concepts such as ‘man’ or ‘woman’ (*Crawford, Paglen 2019*). This assumption leaves little room for hesitations and interpretative complexity. Furthermore, CNNs rely on the assumption that it is possible to capture the visual ‘essence’ in images through statistics, a question that could be critiqued from a philosophical standpoint (ibid.,). CNNs are also “biased towards a distributed, entangled, deeply non-human way of representing the world”, which often makes it difficult to understand how a CNN draws its conclusions (*Offert, Bell 2021*). This has implications for the possibility of providing transparency (i.e., explaining precisely how a CNN makes decisions), as well as abilities to hold the model/researchers accountable for its interpretations.

The VRD does not use machine learning to reproduce semantic binary gender divisions, or output normative textual classifications of objects and humans that appear in images. However, this does not mean that the toolkit escapes the general biases that are associated with CNNs. For instance, there is every reason to assume that the VRD will be better at recognizing and processing certain kinds of content above others – because of how its underlying CNNs have been trained and developed. For example, there are good reasons to assume that the toolkit will be best at processing visual imagery that depicts contemporary, commercial, and Western environments.

### Areas of future development

There are several ways in which the VRD toolkit could be developed further. For example, adding more features for analyzing the effects of applied filters (for instance in the form of statistics showing lost matching results) could provide more transparency and help in finding the right parameters for each filter. More efforts could also go into implementing features for preparing datasets, for instance by removing or ignoring frames that contain a lot of text or single-color (or near-single-color) frames. One possible way of removing text-heavy frames could involve using the OCR features that are embedded in ffmpeg. Regarding the removal of (near-)single-color frames, we have experimented with an algorithm that can currently be found in the is\_monochrome function in the file “image\_preprocessing.py” found on Github (link omitted in this article version to maintain anonymity).

Another way of dealing with the problem of unwanted frames could involve making use of the neural network’s capacities to recognize text, talking heads, and black frames when preparing datasets. In particular, it would be useful if such filters could be applied *before* frames are processed with CNNs and Faiss, as this would likely reduce the image processing time and improve the accuracy of the distance metric distribution. At the moment, however, none of these features have been fully embedded in the VRD.

To further improve the toolkit, we have also played with the idea of using an ensemble method. This involves allowing multiple deep-learning models to identify similar frames, and reaching a concensus before presenting the results. This could help alleviate any issues where a specific model is vulnerable to identifying similar (but not identical) scenes as being the same, for example the previously mentioned talking heads. It would also give us an additional metric to use to express confidence in the results, and automatically flag low-confidence matches. In cases where a specific match is under doubt, an alternative method could be used if it is more accurate, even if this hypothetical method is computationally expensive.

Currently, the VRD is equipped with several features for double-checking, scrutinizing, and handling false positives – that is, instances when sequences have been found and given low distance metric values, even though they do not exemplify an actual instance of reuse. As previously discussed, this may for example concern sequences containing text, black frames, or talking heads. However, the toolkit does not include any features for scrutinizing and handling false negatives – i.e. instances when a correct sequence has not been found and is thus excluded from the matching results. For instance, false negatives may appear if a video contains lots of still images or slow visual movements, which can cause correct matches to be “pushed out” of the top 250 closest neighbors for each analysed frame (see discussion in section [Step 3. Calculate similarity neighbours](#step-3-calculate-similarity-neighbours). Adding features for double-checking false negatives is thus another possible area of future development.

So is improving the database structure that underlies the toolkit, which could be enhanced to better handle multiple fingerprints from the same set of images. We also believe that much could be done regarding the structured and large-scale testing of the performance of the toolkit, for example based on the use of different CNNs, different kinds of source materials, and different tools for calculating nearest similarity neighbors. Given the possibility to embed complex plots, graphs, tables, and visualizations in Jupyter Notebooks, there are also many opportunities to develop how matching results are shown. In many cases, however, these solutions need to be customized to the specific dataset and research questions that are explored. For instance, the possibility to present matching results in graphic visualizations in a relevant way is heavily dependent on the size of datasets, which makes it difficult to embed general features in toolkits such as the VRD. One of the great advantages of Jupyter Notebooks, however, is that users have every opportunity to add and remove features from the VRD toolkit by themselves. A good place to start is by using the [Plotly](https://plotly.com/python/) library, which is currently embedded in the VRD.

Areas of future development aside, we hope that this article has demonstrated the potential of using machine learning to explore audiovisual reuse and showcased the possibilities of using the VRD to navigate historic archives in new ways.

## Bibliography

Appadurai, A.. “Introduction: Commodities and the Politics of Value”. The Social Life of Things: Commodities in Cultural Perspective, edited by Arjun Appadurai, Cambridge University Press.

Arnold, T.and L. Tilton. “Distant Viewing: Analyzing Large Visual Corpora”. Digital Scholarship in the Humanities, vol. 34, https://doi.org/10.1093/digitalsh/fqz013.

Arnold, T., et al.. “Automatic Identification and Classification of Portraits in a Corpus of Historical Photographs”. Proceedings of the CHR 2022: Computational Humanities Research Conference, Antwerp, Belgium. December 12–14., pp. 25–35, https://ceur-ws.org/Vol-3290/short\_paper5571.pdf.

Assmann, A.. “Texts, Traces, Trash: The Changing Media of Cultural Memory”. Representations, vol. 56, pp. 123–34, https://doi.org/10.2307/2928711.

Barbara Flueckiger, .. “A Digital Humanities Approach to Film Colors”. The Moving Image: The Journal of the Association of Moving Image Archivists, vol. 17, no. 2, p. 71, https://doi.org/10.5749/movingimage.17.2.0071.

Bolter, J. D.and R. Grusin. Remediation: Understanding New Media. MIT Press.

Bowker, G. C.and S. Leigh-Star. Sorting Things Out: Classification and Its Consequences. MIT Press.

Bowker, G. C.. “Data Flakes: An Afterword to”raw Data Is an Oxymoron"”. Raw Data Is an Oxymoron, edited by Lisa Gitelman, MIT Press, pp. 167–72.

Crawford, K.and T. Paglen. “Excavating AI: The Politics of Images in Machine Learning Training Sets”. AI & SOCIETY, https://doi.org/10.1007/s00146-021-01162-8.

De Kosnik, A.. Rogue Archives: Digital Cultural Memory and Media Fandom. The MIT Press.

Delfieu, T., et al.. Snoop Video. https://www.culture.gouv.fr/Media/Medias-creation-rapide/Snoop-2022-03-22.pdf.

Denton, E., et al.. “On the Genealogy of Machine Learning Datasets: A Critical History of Imagenet”. Big Data & Society, vol. 8, no. 2, p. 205395172110359, https://doi.org/10.1177/20539517211035955.

DeVries, T., et al.. Does Object Recognition Work for Everyone?. arXiv, http://arxiv.org/abs/1906.02659.

Ernst, W.. “The Media Epistemic Value of Sonic Analytics Tools. A Commentary”. Internet Histories, vol. 5, no. 1, pp. 48–56, https://doi.org/10.1080/24701475.2020.1862528.

Google, .. How Google Fights Piracy. Report, https://storage.googleapis.com/gweb-uniblog-publish-prod/documents/How\_Google\_Fights\_Piracy\_2018.pdf.

Greenblatt, S.. Shakespearean Negotiations: The Circulation of Social Energy in Renaissance England. University of California Press.

Holdsworth, A.. Television, Memory and Nostalgia. Palgrave Macmillan UK, http://link.springer.com/10.1057/9780230347977.

Arnold and Tilton - 2019 - Distant Viewing Analyzing Large Visual Corpora.pdf. Arnold and Tilton - 2019 - Distant Viewing Analyzing Large Visual Corpora.pdf.

Jofre, A., et al.. “What’s in a Face? Gender Representation of Faces in Time, 1940s-1990s”. Journal of Cultural Analytics, vol. 5, no. 1, https://doi.org/10.22148/001c.12266.

Kirschenbaum, M. G., et al.. Digital Forensics and Born-digital Content in Cultural Heritage Collections. Council on Library and Information Resources.

Kirschenbaum, M. G.. Mechanisms: New Media and the Forensic Imagination. MIT Press.

Kopytoff, I.. “The Social Life of Things: Commodities in Cultural Perspective”. The Social Life of Things: Commodities in Cultural Perspective, edited by Arjun Appadurai, Cambridge University Press, pp. 64–94.

Lessig, L.. Remix: Making Art and Commerce Thrive in the Hybrid Economy. The Penguin Press.

Macdonald, M.. “Performing Memory on Television: Documentary and the 1960s”. Screen, vol. 47, no. 3, pp. 327–45, https://doi.org/10.1093/screen/hjl025.

Müller-Budack, E., et al.. “Geolocation Estimation of Photos Using a Hierarchical Model and Scene Classification”. Computer Vision – ECCV 2018, edited by Vittorio Ferrari et al., vol. 11216, Springer International Publishing, pp. 575–92, https://link.springer.com/10.1007/978-3-030-01258-8\_35.

Müller-Budack, E., et al.. Finding Person Relations in Image Data of the Internet Archive. pp. 229–40, http://arxiv.org/abs/1806.08246.

Müller-Budack, E., et al.. “Ontology-driven Event Type Classification in Images”. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, pp. 2927–37, https://doi.org/10.1109/WACV48630.2021.00297.

Moretti, F.. Distant Reading. Verso.

Navas, E.. Remix Theory: The Aesthetics of Sampling. Springer.

Offert, F.and P. Bell. “Perceptual Bias and Technical Metapictures: Critical Machine Vision as a Humanities Challenge”. AI & SOCIETY, vol. 36, no. 4, pp. 1133–44, https://doi.org/10.1007/s00146-020-01058-z.

Olsson, J.. The Life and Afterlife of Swedish Biograph: From Commercial Circulation to Archival Practices. The University of Wisconsin Press.

Pereira, G.and B. Moreschi. “Artificial Intelligence and Institutional Critique 2.0: Unexpected Ways of Seeing with Computer Vision”. AI & SOCIETY, vol. 36, no. 4, pp. 1201–23, https://doi.org/10.1007/s00146-020-01059-y.

Saleh, L.and N. van Noord. “The Computational Memorability of Iconic Images”. CHR 2022: Computational Humanities Research Conference, Antwerp, Belgium.

Tan, M.and Q. V. Le. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv, http://arxiv.org/abs/1905.11946.

Taurino, G.and M. Boni. “Maps, Distant Reading and the Internet Movie Database: New Approaches for the Analysis of Large-scale Datasets in Television Studies”. VIEW Journal of European Television History and Culture, vol. 7, no. 14, p. 24, https://doi.org/10.18146/2213-0969.2018.jethc151.

Tubaro, P., et al.. “The Trainer, the Verifier, the Imitator: Three Ways in Which Human Platform Workers Support Artificial Intelligence”. Big Data & Society, vol. 7, no. 1, p. 205395172091977, https://doi.org/10.1177/2053951720919776.

van Es, K., et al.. “Tool Criticism and the Computational Turn. A “methodological Moment” in Media and Communication Studies”. Medien & Kommunikationswissenschaft, vol. 69, no. 1, pp. 46–64, https://doi.org/10.5771/1615-634X-2021-1-46.

Wales, A.. “Television as History: History as Television”. Television and Criticism, edited by Solange Davin and Rhona Jackson, Intellect.

Wevers, M.and T. Smits. “The Visual Digital Turn: Using Neural Networks to Study Historical Images”. Digital Scholarship in the Humanities, vol. 35, no. 1, pp. 194–207, https://doi.org/10.1093/llc/fqy085.

Wevers, Melvin, .and T. Smits. “Detecting Faces, Visual Medium Types, and Gender in Historical Advertisements, 1950–1995”. Computer Vision – ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II, edited by Adrien Bartoli and Andrea Fusiello, vol. 12536, Springer International Publishing, https://link.springer.com/10.1007/978-3-030-66096-3.

Wevers, M., et al.. “What to Do with 2.000.000 Historical Press Photos? the Challenges and Opportunities of Applying a Scene Detection Algorithm to a Digitised Press Photo Collection”. TMG Journal for Media History, vol. 25, no. 1, p. 1, https://doi.org/10.18146/tmg.815.