

---

*Special Issue: Critical Technical Practice(s) in Digital Research***(De)constructing machines as critical technical practice****Winnie Soon      and Pablo R Velasco****Abstract**

*This paper discusses the role of technology under the framework of Critical Technical Practice specifically in the form of constructing artefacts and deconstructing tools in order to produce what Philip Agre would describe as ‘reflexive work of critique’ (Agre, 1997:155). By presenting the activities and methods used in the teaching and shaping of undergraduate courses, this paper aims to show how technical objects, such as data, datasets, application programming interfaces and machine learning models, can be considered as discursive subjects, demonstrating pedagogical understanding across fields. The courses operate in the humanities tradition and take critical technical practice as a didactic approach, insofar as software and data are understood and manipulated on an instrumental level, while encouraging critical engagement and embodied reflection that bridge the technical and social/cultural domains. Within this pedagogical approach, critical is not only understood as a paradigm of rationality or quantitative, data-driven argumentation, but as adopting a critical position – that is, to research and reflect on the social structures and cultural phenomena entangled with digital objects, bodies, tools, methods and software production. By embracing work-in-progress and reflexive exploration, we aim to extend the notion of critical technical practice by unfolding how (de)constructing machines can be achieved beyond thinking of technology as neutral instrumentalisation. The challenge is how to find a balance, not only as researchers but as educators, unfolding aspects of both formality and functionality as well as questioning and understanding technology at a discursive and critical level. We argue that learning technical practice in an educational setting is not an end, but rather a means to question existing technological structures and create further changes in socio-technical systems.*

**Keywords**

Aesthetic programming, application programming interfaces, critical technical practice, data, datasets, data studies, machine learning models, pedagogy

Aarhus Universitet, Aarhus, Denmark

**Corresponding author(s):**

Winnie Soon, Aarhus Universitet, Aarhus 8000, Denmark. Email: [wsoon@cc.au.dk](mailto:wsoon@cc.au.dk)

**Introduction**

This paper arises from the challenges of teaching, learning and working with technical objects in higher education curricula, specifically the understanding of data beyond traditional science-related disciplines.

Increasing attention is being paid to understanding how data is ubiquitously captured, operated and connected both technically and culturally, and how such data processes (re)configure the way we see, perceive and know things beyond technical configurations. It involves micro and macro decisions, and hence design assumptions that require a critical lens to deconstruct what Philip Agre might term the ‘grammars of action’ (Agre, 1994). This refers to the ‘tacit assumptions’ within technology that actively structure ‘human activities’ and narrate the perceived reality. In this way, programmable software and data organisation *model* our daily activities beyond mathematical information with social and cultural implications. Therefore, data should be considered a socio-technical object that requires comprehending it both critically and technically. But how should such complexity be introduced pedagogically?

A critical approach to the study of data as technical objects presents the challenge of reflecting on the complexity beyond – yet within – their technical elements, while accounting for their inherent materialities and enacting practices. This paper discusses the teaching and learning of such data objects under the framework of Critical Technical Practice (CTP) specifically in the form of constructing artefacts and deconstructing tools in order to produce what Agre would describe as ‘reflexive work of critique’ (Agre, 1997: 155). We use the term ‘machine’ to refer to the types of objects prone to (de)construction both due to the relational nature, and due to their technical elements. This umbrella term allows us to unfold dissimilar technical objects, while preserving the main characteristic of them working within a broader machinery. The term is useful to dissect both the material parts of technical objects, and the ‘assemblages that include both humans and tools’ (Johnston, 2008: 200). Software, databases or computational processes can all then be understood as ‘machines’ prone to analytic (de)construction and making practices.

This paper presents three inter-related technical objects: dataset, application programming interface (API) and machine learning model, which we have used in the teaching and shaping of our undergraduate courses. All too common in data science and data-oriented humanities syllabi, these three interdependent objects are key to understanding state-of-the-art implementations and the contemporary impact of technology. However, they are usually addressed either as practical tools or as objects of critique. Our approach follows a CTP framework that aims to bridge this gap, by focusing on their technical understanding as a point of departure for critique. The aim is to show how technical objects can be considered as discursive subjects, bridging pedagogical understanding across fields.

Our approach to critical technical practice allows us to comprehend the aforementioned complexity of these technical objects alongside their technical usage. We refer to this process as *(de)construction*<sup>1</sup>. This term, on the one hand, encloses the dissection and close interrogation of specific technical elements (e.g. the *index* of the dataset, or the *query* of the API) as points of departure to examine the larger aesthetic, cultural and political effects of their material machinery and agency. On the other hand, the critical aspect of our work does not exist in a theoretical vacuum, but together with the technical construction of these objects. That is, we also follow the motto that criticality happens with and through construction. This also references the genre of critical making that describes the importance of material engagement and building prototypes as a form of research practice (Ratto, 2011; Ratto et al., 2014). But beyond constructing artefacts, the following sections will show examples of the (de)construction of tools as well. As such, our pedagogical approach is to a large extent invested in the technical re-construction of these objects beyond the prism of instrumentality.

This article is primarily based on the two courses, Aesthetic Programming (AP) and Data Studies (DS), which are operated in the humanities tradition<sup>2</sup>, adapting Critical Technical Practice as a didactic approach. More specifically, software and data are understood and manipulated on an instrumental level while

encouraging critical engagement and embodied reflection (Kinsella, 2007) that bridge both the technical and the social/cultural domains (Gollihue and Browning, 1988; Dourish et al., 2004; Harwood, 2019; Ratto et al., 2014; D' Ignazio and Klein, 2020). Within this pedagogical approach, critical is not only understood as a paradigm of rationality (Siegel, 2017) or quantitative, data-driven argumentation (Holmes et al., 2015), but as adopting a critical position –i.e. to research and reflect on the social structures and cultural phenomena entangled with digital objects, bodies, tools, methods and software production.

By embracing work-in-progress and reflexive exploration, we aim to extend the notion of critical technical practice by unfolding how (de)constructing machines can be achieved beyond thinking of technology as neutral instrumentalisation. As humanities scholars who are situating into the fields of Software and Data Studies, we found that the challenge is how to find a balance, not only as researchers but as educators, unfolding aspects of both formality and functionality as well as questioning and understanding technology at a discursive and critical level. We argue that learning technical practice in an educational setting is not an end, but rather a means to question existing technological structures and stimulate intervention in socio-technical systems. Below, we explore how three technical objects – the dataset, APIs and machine learning models-are pedagogically (de)constructed. This article does not aim to provide detailed pedagogical toolkits for adoption because our examples have been selected, highlighted and put together out of the entire two 13 weeks full courses, in which other pedagogical elements are considered, such as learning outcomes, levels of students and how things and individual classes relate to one another. Instead, our three selected technical objects serve the purpose of unfolding our notion of (de)construction as an expansion of critical technical practice.

### *The dataset in practice*

The database, as a concept and an object, has become a pervasive condition of possibility for the life and organisation of data. Usually relegated to the background (*backend*), these ubiquitous objects mediate almost every digital interaction. Either as minimal datasets – any structured collection of data records- or as aggregations of the former in massive databases – for example, the World Data Centre for Climate-these objects function as storage and schemata for almost any digital data. The techniques related to our examples in this contribution (capturing, managing, querying, etc.) are in one way or another linked to (and dependent on) some kinds of dataset. As such, a CTP approach entails not only shedding light on the relevance of these digital objects, but also exploring how this very pervasiveness is embedded in and defines broader biopolitical and social structures, both as a framework or narrative, and as a baseline for novel production of knowledge. In particular, the (de)construction of a dataset as a technical object paves the way for a critical reflection on its “site of production” (Latour, 2017: 117), or the design space of this object, usually reserved to the data engineer.

Minimally defined as a structured collection of data or an organised collection of data stored as multiple datasets, the database also depicts an ontology of the world in computational terms in conjunction with algorithms and data processing (Manovich, 2001). Dourish (2014) extends this line of research on the database, focusing more on its characteristics as an infrastructure and less as media. Like Manovich, he stresses the materiality of information in this format, and considers it a fundamental characteristic of the digital. According to Dourish, the digital is inherently material, and this materiality is not simply based on the structure being represented as electrical or magnetic traces, that is, it is not only based on its physical

infrastructure. It is also based on the fact that the material ‘constraints’ allow the specific infrastructure to work in the form of a database, that is, the way in which information is structured and coded for specific executions. How information is encoded into databases is particularly relevant insofar as these devices perform a crucial role in the categorisation of subjects: populations, governmental decision-making, statistical findings, personas and other macrostructural proceedings are moulded by the specificity of the database design and administration (Cheney-Lippold, 2017; Goriunova, 2019; Hacking, 1986). Such entering and recording of data do not simply represent reality but also enact it, as Ruppert puts it: ‘each entry is an inscription and the result of specific operations that mediate, translate, summarise and materialise the sets of relations that make up each context’ (2013). The encoding practices in the creation of datasets are thus not sterile or raw bits of existing information merely represented in technical schemata, but operations that continuously *enact* reality within a computationally mediated structure.

A crucial part of our pedagogical experiment on the reformulation of data representations consists in (de)construction through practice. That is, undoing the implicit encoding practices of a technical object by critically re-making it. The praxis of the courses, thus, retains a certain level of instrumentality, which is not an end in itself but a means of reflection (Dourish et al., 2004). Unlike more practically oriented projects, this approach does not necessarily entail the production of a whole or final database, but does require:

- a. An understanding of the subject’s role in relation to datasets/databases,
- b. Limitations as to what counts as data in compliance with data frameworks and
- c. Identification of the politics of data embedded in such a technical object.

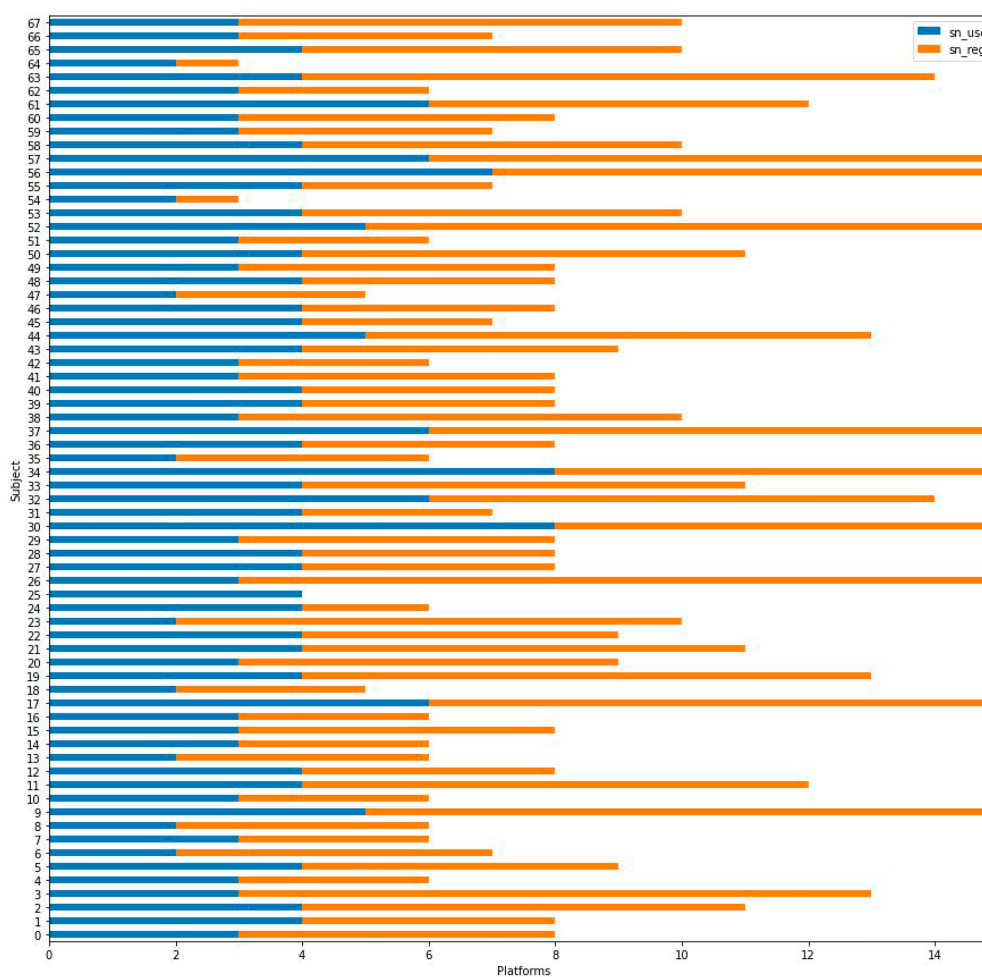
As such, (de)construction of datasets in the classroom starts by positioning the role of the student as both the object and the subject of data, and questioning the scope of its agential qualities. Such entanglement is what Karen Barad would call *phenomena* or ‘the ontological inseparability/entanglement of intra-acting agencies’ (2012: 139). The study of technical objects in our courses follows Barad (Barad, 2007; Barad and Tuin, 2012), insofar as we recognise agency not only as properties (of subjects or things) but as a process (an *enactment*) of entanglement between different entities (e.g. subjects and their data representations), which has the potential of reconfiguring such entanglements. Our (de)constructing process starts by signalling this entanglement through the lens of a dataset.

### *The subject embeddedness in datasets*

At the very beginning of the course, for example, a dataset is built based on information provided by the cohort. The survey is designed in a deliberately bounded way to make evident the distressing factors involved in capturing data (e.g. a non-skippable binary input on gender). This dataset is then presented in different common data visualisation forms (e.g. bar charts, scatter plots, network graphs, geolocated maps, etc.), which are usually familiar to students and, sometimes unconsciously, embedded with a presumption of scientific, objective knowledge. The arguably skewed, yet ‘raw’ (Gitelman, 2013) and ‘objective’, data in most cases clashes with students’ own self-representations, and opens the space for an engaged discussion about the cultural politics of data structures. Figure 1 shows, as an example, what is perhaps one of the most familiar representations of ‘raw’ data (Drucker, 2014), the bar chart. The data shown in this graph, however, comes directly from the students surveyed information and, as such, is subjected to a relational and subjective examination on the part of the students. Contrasting the arguably explicitly objective information encoded in this visual representation with the more nuanced embodied life of data, helps to position the



malleability of data processing practices, and attunes the critical eye of the student for any claims of data objectivity that ignore or minimise situated practices (D' Ignazio and Klein, 2020; Haraway, 1988).



**Figure 1.** Bar graph showing the number of platforms used (blue) and installed (orange) by students (python's matplotlib library).

Although when it remains on the purely analytical level, this exercise proves to be useful for the forthcoming implementation and management of datasets (in and outside the course), as it makes evident: a) the importance of database design and information encoding in the configuration of political assemblages (Dourish, 2014; Latour, 2007), b) the close relation between established data representations and knowledge claims (Drucker, 2014; McNerny et al., 2018) and c) that a majority of the so-called big data is populated with or by individuals (Uprichard, 2013). This approach attempts to reduce the gap between the quotidian and embodied 'authentic experience' (Agre, 1997: 146) of the students and the abstract contents of a digital dataset. The vernacular reading of datafied models tends to disconnect associated conceptual interpretations from their day to day material experiences. Critical Making workshops are in part a response to re-connecting 'deterministic, conceptual understandings of the role of technology in social life, and the more material and nuanced understanding of how one relates to them' (Ratto, 2011: 253). This is not different in the classroom, and remains a challenge shared by didactical approaches seeking to materialise critical aspects within instrumental techniques and objects (Harwood, 2019; Ratto, 2011; Rogers, 2019; Soon and Cox, 2020).

The relation between subjectivity and data is not only underlined for students as passive individuals, but also primarily as data-practitioners, i.e. subjects involved in the creation and design of data-driven classification. While engaging with tools, techniques and libraries that are commonly used in data science practices, the role of the student is modified as it is articulated within the gaze of the data scientist and the data engineer (Beer, 2018; Foucault, 1978). Both personas, or characters involved in the database construction, the data manager and the data structure designer are enacted by students as they start to manipulate information in accordance with the standard practices of data science: cleaning data, identifying patterns, removing outliers and aggregating parameters as well as other quantitative analysis procedures. According to Beer, both personas are understood as qualified experts in the data imaginary due to their skills: analytics and logic, respectively. The gaze in such roles is not merely instrumental, however, insofar as these personas are used as prisms for a critical approach, the technical expertise not only allows the students to identify patterns (work *with* data), but also to know ‘which patterns are seen to matter’ (Beer, 2018: 123), i.e. stressing the influential role of these personas in steering what is to be understood as knowledge. The design and manipulation of a dataset comes with an enhanced sense of agency (the potential for entanglement reconfigurations) and a mode of knowledge embodied in the aforementioned personas (Beer, 2018; Kitchin, 2014). As the datafication of society is intensified (Dijck, 2014; Mejias and Couldry, 2019), the data practitioner seems to have an advantageous position to visualise, adapt and generate representations of reality.

## Design and technical schemata as rules for inclusion

Collected, captured or generated data, regardless of its origin, adapts to a pre-designed dataset structure and pre-existing assumptions when using computational methods. The initial survey exemplified above is also useful to emphasise the influence of the student as data practitioner on possible processing outcomes. Collecting data through a binary model and structuring it into discrete categories, for example, clearly limits the possible readings of a dataset. In these kinds of surveys, students are intentionally asked two binary-encoded questions without the possibility of a third option (Figure 2). The purpose of this exercise is to underline how the design of possible data entries in a dataset inevitably snowballs into further assumptions, and questionable correlations. The aim of these depictions of data is not only to highlight the importance of a careful design in the process of data collection (Passi and Jackson, 2017), but also, and mainly, to emphasise that technical objects like datasets come with their own rules for what can or cannot be included in them. Collection devices do not measure the properties of independent objects, but instead observe the entanglement of an apparatus of observation and the observed (Barad and Tuin, 2012). That is, computational practices for capturing, managing and visualising have built-in epistemologies (Marres and Weltevrede, 2013; Ruppert et al., 2013), like the mentioned binary and discrete entries, that guide and model further knowledge claims and thus shape possible ontologies.





**Figure 2.** Alluvial diagram showing category correlation of two (exclusive) binary entries in a survey (visualisation made using RAWGraphs, see Mauri et al., 2017).

Dourish identifies a relation between the organisations behind the design of data structures and the organisational interests of those in charge of the development. He argues that the development of *System R* (an old IBM system), a relational data model that was the blueprint of many contemporary database systems, was an overall design not only related to data structures, but also to a business model to offer computing services and computing architecture manufacturing. Dourish argues that IBM's position allowed the corporation 'to develop its computer architectures to enhance performance in executing relational database transactions, and to define the benchmarks and measures by which database systems would be evaluated' (Dourish, 2014: 15). The grammar of the relational database, in this case, is related to sociomaterial configurations that involve a business model, software systems, and physical manufacturing of an organisation. Indeed, the capture process identified by Agre is never completely technical, and it includes elements of interpretation, strategy and institutional dynamics:

"capture is never purely technical but always sociotechnical in nature. If a computer system 'works' then what is working is a larger sociopolitical structure, not just a technical system (...) if the capture process is guided by some notion of the 'discovery' of a pre-existing grammar, then this notion and its functioning should be understood in political terms as an ideology" (Agre, 1994: 748).

A (de)constructive approach helps to unveil the grammars of the dataset by identifying unitary actions, such as a gender-based input in a boolean datatype, and means to combine such actions into a sequence, e.g. by correlating binary variables. Grammars of action are explained by Agre mostly in relation to AI systems' development during the 80s, however, they are meant to be understood in a general sense as the construction of "systematic representations of organisational activities" (Agre, 1994: 745). The meticulous observation of such grammars is an attempt to embed a critical practice within technically design processes. While the former binary example, in terms of both gender conceptions and data structures, is mostly related to the design process in the data collection (i.e. formulating the survey), the inclusion of data through this and other means like automated collection (for example, APIs as illustrated below) is bounded by the possible schemata, or grammar, of a dataset. A critical observation of such schemata shows that, while most information is potentially datafiable, some data is more fitting than other data. An immediate example of 'fitting' information is data susceptible to quantification, discreteness, clarity, etc. A suitable enacting of critical technical practices should aim to critically show this data 'compliance' while working with datasets.

In our classroom settings the production of the dataset goes hand in hand with the understanding, and possibility for manipulation, of data structures through ready-to-use tools like *Openrefine*, suitable data notation like the json<sup>3</sup> data format type or dedicated management libraries such as python's *Pandas*. Working with *Dataframes* for example, a canonical Pandas dataset within data science helps to understand the structural elements of the dataset category (e.g. the essential role of an index, regardless of its contents, or the entry design), as well as the tubular representation of objects within such a model. Questioning the basic elements of a dataset structure in practice makes it possible to reveal the strict epistemologies attached to this object, which are otherwise invisible in their output (e.g. a 'result' from a website return call).

The creation of a *Dataframe* (see Figure 3) will generate an index as an essential element to manage the contents of such a table, even if the data was imported from an unstructured source (e.g. a spreadsheet or a non-indexed table). Students are asked to generate datasets either with web scraping tools or manually. The difference between using these two data capturing methods is not negligible, as the (already) machine-readable data is separated from the to-be-made machine-readable data. However, regardless of its origin, the Pandas library adapts the data to an indexed and tabular format, revealing the necessary adequation of any data in the dataset scheme. Further cleaning of the dataset – a common data science practice consisting of distilling valid, accurate, complete and machine-readable data-reveals the types and formats of information that can be included in the ontology of the dataset. The inclusion and adaptation of dissimilar data into valid formats is made following common technical protocols, thus providing the student with the technical skills associated with data manipulation. The learning goals of our approach, however, are focused on questioning the epistemological and ontological nuance built within these technical practices. Such a critical technical approach can be applied to the study of other technical objects. That is, the technical skills are not an end in themselves, but a means to critically analyse the entanglement of sociomaterial practices that come with them.

Importing a (dirty) dataset and inspecting it

```
[2]: df = pd.read_csv("dirtydata.csv") # the file should be in the same folder as this notebook, on your machine
```

This is a 'dirty' dataset exported directly from the dataminer chrome extension. Let's quickly orient our selves of the 'info':

```
[3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38 entries, 0 to 37
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Beskrivelse  38 non-null      object
1   Pris         38 non-null      object
2   Hvor         38 non-null      object
3   Billede      38 non-null      object
4   Column 5     38 non-null      object
5   Active       38 non-null      int64
dtypes: int64(1), object(5)
memory usage: 1.9+ KB
```

Let's inspect a few of the rows of the data as is:

```
[4]: df.head(5)
```

	Beskrivelse	Pris	Hvor	Billede	Column 5	Active
0	Herracer, andet mærke Motobecane, 50 cm stel...	3.000 kr.	2200 København N	https://i.ebayimg.com/images/g/5mAAOSwXmRIT3s...	I dag	1
1	Herracer, andet mærke, Vi kan altid åbne eft...	1.500 kr.	8260 Viby J	https://i.ebayimg.com/images/g/ZSsAAOSwlcFIT1s...	I dag	1
2	Herracer, Cipollini, Chipolini RB1000 str. ...	50.000 kr.	2930 Klampenborg	https://i.ebayimg.com/images/g/e9wAAOSw3iNITp6...	I går	0
3	Herracer, FOCUS Izalco Max 9.7 Disc , 56 cm ...	34.000 kr.	2600 Glostrup	https://i.ebayimg.com/images/g/Bj8AAOSwA15ITpF...	I går	0
4	Herracer, BMC TMR01, 58 cm stel, BMC timemac...	16.900 kr.	2650 Hvidovre	https://i.ebayimg.com/images/g/-koAAOSw5xZToT...	I går	0

The column "Column 5" seems a bit out of place, right? Let's rename it to something more representative!

**Figure 3.** Using Jupyter notebooks and the Pandas library to clean a 'dirty' dataset captured using web scraping (python's pandas library).

## API as technical objects

Building upon the knowledge of data notation and format in datasets, in this section we examine an Application Programming Interface (API) to further explore other forms of implicit encoding practices that are connected to wider internet networks that are less visible to end users. An Application Programming Interface (API) is considered a technical object in computing, in which the corresponding specification and protocols determine the relations and interactions between software (Cramer and Fuller, 2008: 149). Unlike a 'user interface' in which a user can easily interact with a graphical manifestation, an API is not meant to communicate between users and machines, but within machines and software programs. Amazon, for

example, the largest marketplace in the world, implements most of their software code in the form of APIs to communicate with other code (Yegge, 2011). But APIs do not just operate within their own company workflows, but also on the Internet across continents.

Web-based APIs specify the type of an API that is operated within Internet network protocols, which are commonly offered by software or platform providers nowadays to provide access to their data, such as Facebook, Twitter, Weibo, Google Search and Google Maps, among others, allowing third-party programs to either interact with and build upon the corresponding platforms for wider usage (as in the case of Twitter bots that push data onto the platform by means of programs), or extract data for further application development or analysis. With the increased performance and capacity of cloud computing, data storage and processing, we can also observe how software changes from packages to software as services (Haigh, 2002; Gurses and van Hoboken, 2017), or we may even say APIs as services (Soon and Pritchard, 2021). But why is this a relevant and important object to discuss in the humanities and social sciences? Why does it matter to us?

If we narrow down web APIs to a form of data query, then it is about what data services are being provided to the public that matter to us. This service offering is conditioned by terms and conditions, specification (technical schemata), legal liability, design decisions, forms of interactions (Seaver, 2019) in order to govern the querying of data. How is data requested via web APIs? What is the data available? How does data return to the other side upon request? How is data being structured and represented? These are the questions we start to open the discursive space in a classroom about the process of data transmission and communication. In other words, API is a form of two-way communication process related to information processing that is coupled with data selection, extraction, transmission, and presentation through the logic of request and response between programs (Soon, 2016: 124; Snodgrass and Soon, 2019; Soon and Cox, 2020; Sollfrank and Soon, 2021). Such a logic may not be as apparent as other graphic-driven interfaces. However, APIs have been running actively within corporations (Gurses and van Hoboken, 2017), smart cities (Raetzsch et al., 2019), Internet platforms (Snodgrass and Soon, 2019; Burkhardt et al., 2020) and many other software services. Indeed, APIs are quite digitally ubiquitous, yet not user-centred or graphic-driven, making them and their logic a particularly interesting object of research to get closer to the less apparent infrastructure. Beyond how APIs can be used for further development in a business setting or software production, the challenge is how to (de)construct the complexity of such socio-technical objects as part of critical making. The question then becomes how to explore the interwoven social and technical aspects of APIs (materially) in order to open up the critical and discursive space to designers, artists, researchers and students for reflexive making.

## **APIs as technical-discursive objects**

Educators, practitioners and scholars have started to incorporate APIs to extend the technical nature into curriculums and other knowledge-based development within arts and humanities.<sup>5</sup> In the broad field of data studies, the focus on digital objects consists of new techniques to manage and organise data. According to Hui, '[t]he significance of the new technique of data processing we now call the digital is not only that with computers we can process large amounts of data but also that by operating with data the system can establish connections and form a network of data that extends from platform to platform, database to database'. (Hui, 2012: 388) In the same way as with the dataset example above, this perspective of digital objects shifts the focus of data from its discreteness to material relations, suggesting that data is produced with the priority to

facilitate connections and portability, which is an important part of query data in web APIs.

Beyond the two-way communication protocols and connections with other entities, another key element of web APIs is their real-timeness. Many web APIs offer real-time query, meaning that each data request may contain different results even with the same given parameters. This temporal perspective resonates with the scale and frequency of user-generated content that is distributed via networks as endless updates, but it also points to the dynamic of algorithms that render data (in)visible – what data prioritises to be displayed and seen from a massive database. These temporal updates and spatial rendering are made possible by digital techniques in which data ‘are storable, traceable and analysable...that [are] kept in databases’ (Soon, 2016: 122). Querying data is about retrieving and navigating databases with the kinds of data that one would like to draw upon. Not only the design and construction of databases are permeated with built-in epistemologies as argued before, but through real-time queries APIs may also enact different worldviews.

Each query presents a customised view on what the world is represented at the moment which is unpredictable (ibid). This matters to us because it changes our experience of the world and the perception of reality. Although it is commonly understood that an API is a technical object to facilitate connectivity, data analysis and exchanges, as well as to solve many interoperability issues (Asano et al., 2016; Snodgrass and Soon, 2019), we also see it as a discursive object, opening up for discursive reflection beyond technical fix. This can be traced back to Foucault’s notion of discourse, revealing power relations that contingently produce knowledge via explicit and implicit rules and categories (Foucault, 1971). In other words, API is also a material object that is constructed with assumptions, beliefs, judgements, organising social orders and operating as culture machines (Finn, 2017) in a border sense. Scholars who research on technologies use specifically the notions of ‘discursive turn to AI’ (Katzenbach, 2021) and ‘discursive formations’ (Bucher, 2013) to questioning the representation of certain AI/API knowledge and their technical neutrality.

Through understanding APIs as technical-discursive objects, we see how APIs organise data with algorithms that are relatively hidden from users. *Query the query* becomes more urgent beyond ‘matters of fact’ as ‘matters of concern’ (Latour, 2004: 231) and even ‘matters of care’ (Puig de la Bellacasa, 2017). It becomes a matter of wider societal concern and responsibility especially for profit-driven giant companies and platforms that have orchestrated the majority of Internet media outlets (such as social media), and massive knowledge bases such as Google Search with trillions of searches/queries per year worldwide (Sullivan, 2016). The power to manipulate and organise how we see and perceive the world and reality cannot be underestimated (Finn, 2017; Noble, 2018; Bucher, 2018).

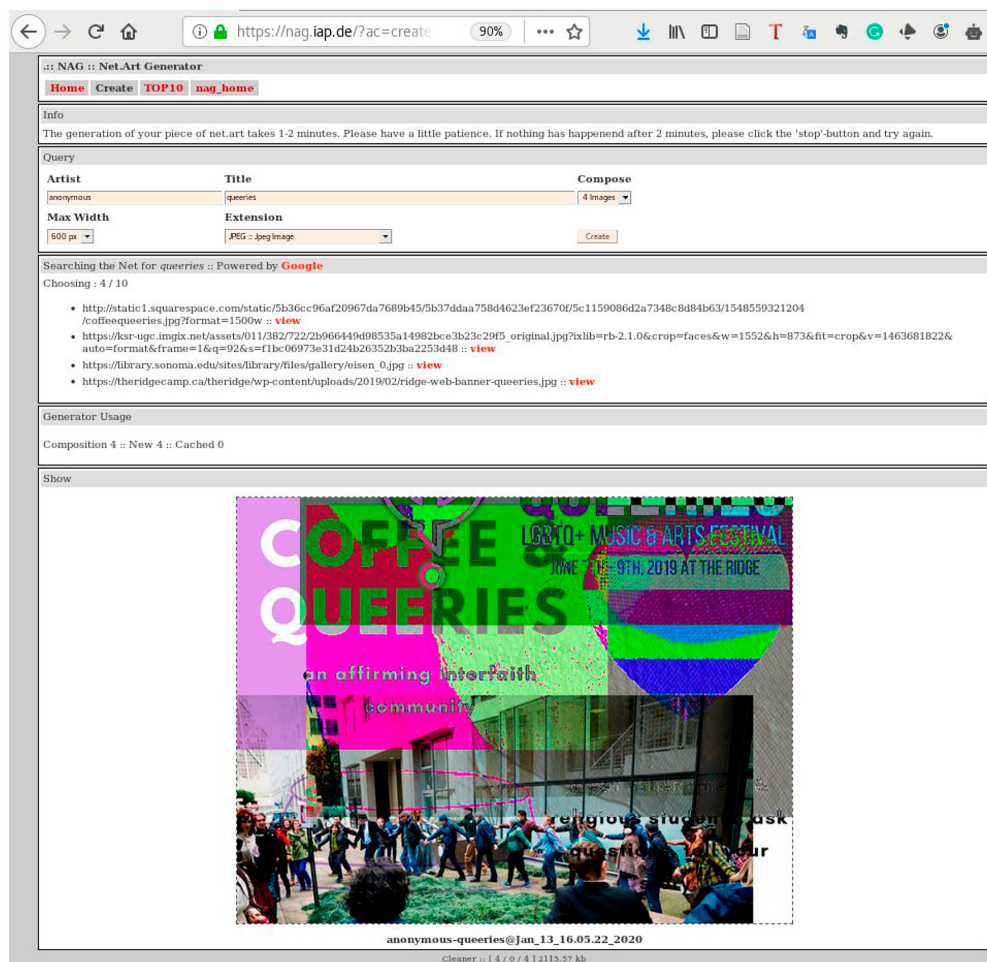
Tellingly, the giant companies that offer web APIs for data retrieval determine the logic of access, for example access in terms of how one could retrieve the data, but often the changes in access are seamless to the users/developers, and that will result in many dependent applications failing to work.<sup>6</sup> Access to databases via web APIs usually requires the study of specification which is written in a particular way (for example, the data format of JSON is organised in the form of name and value pairs) and with grammars of action, a ‘grammar of states and state-changes’ representing organisational activities (Agre, 1994: 749), that facilitate developers to use in programming code with almost any kinds of programming language. The specification is usually written in a straightforward and technical manner, listing the ‘available’ parameters, but rarely disclosing the detailed logic behind, or what is made unavailable (this will be discussed in specific details in the next section). Therefore, it is easy to accept what has been presented in the specification as ‘matters of fact’ (Latour, 2004) that are distanced from your subject and guide the focus on how to implement the parameters instrumentally instead, thereby overlooking the authentication processes, liability



and terms and pricing models for accessing web APIs as part of the standard service provided. As such, the conditions of data production are matters of concern instead.

## net.art generator

One of the (de)construction approaches we have used is to observe how a piece of software incorporates an API. Using software-based artworks to deal with some of the former complex socio-technical issues in a classroom setting is something we found fruitful, especially the software exhibits some forms of interactivity. *Net.art generator* (nag) is an application that runs on a web browser in which a user can input a key term and the program will generate an image collage based on Google Image Search results in real time (see Figure 4), created by artist Cornelia Sollfrank<sup>7</sup> in 1997. One of the classroom exercises is simply to go online, and therefore everyone can play with and explore the web application to consider the processes between inputting the keywords and outputting an image collage. By considering the output images, the students are encouraged to think about how *real-time* images are gathered from different sources which are taken out of context, and how individual images can be combined and applied with visual effects that produce a collage (i.e. the manipulation of data).



**Figure 4.** The net-art generate web interface with the input keyword 'que(e)ries' from the software repository of Aesthetic Programming (Soon and Cox, 2020).

*nag* uses Google Image Search API in the program, which means it needs to communicate with the

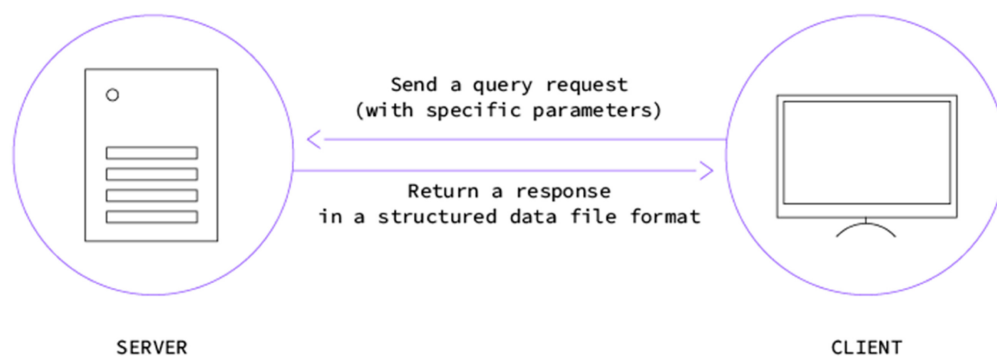
Google server via the ‘request and response’ query logic, in which a program can request images (with certain parameters), and then Google will return search images in a particular data file format (see Figure 4). To fully understand this specific query logic and its implications, a smaller version of *nag* is made to focus only on this part for teaching purposes and easier comprehension. This smaller version then requires students to engage with the process of obtaining the end-to-end flow by first registering the web APIs, then studying the specification, and finally putting the APIs into actions by tinkering with a program as a form of (de)construction

## (De)constructing the web API

In this section, we will detail the three processes of (de)constructing the specific API object by using the smaller version of *nag* as part of critical technical practice, which is both paying attention to the entanglement of matter and meaning: How APIs work and what forms of potential discursive discussion can be posed

### 1. Getting API keys & Knowing the limitations

The Google Image Search API is used as the prime focus to demonstrate the process of registration and authentication, as well as the required terms and conditions for usage. Two important keys are needed: 1) API key 2) Search Engine ID. These are used to identify the unique application that sends requests to the server, and as such, the platform server (ie Google) can identify who is getting the data as well as the corresponding traffic and usage. Before obtaining the two keys, students read through the terms and conditions page to understand the services offered, legal liability and limitations. For example, the keys cannot be transferred to others, and the free version only has 100 free requests, after which additional costs will be incurred. But the pricing model is not that simple, because other Google API services, such as the Google Maps API, have a different charging scheme and free usage. Instead of taking the pricing as simply a piece of factual information, this discrepancy opens up discursive reflection in further questioning about why there is such a difference between services, what the differences are in acquiring (public) data from Google, and what kinds of data are most valuable from a business perspective (Figure 5).



**Figure 5.** Client-server communication from the book titled ‘Fix my Code’ (Sollfrank and Soon, 2021).

### 2. Locating (un)available parameters

After obtaining the two unique keys and other configurations, students can put them into the small program



and run successfully (see Figure 6). The next step would be to tinker with the program code so that a new image will return when a different keyword is used. This further illustrates what other parameters are needed to send a single request to the server (see Figure 7) in which an API request is simply a URL like this: <https://www.googleapis.com/customsearch/v1?key=APIKEY&cx=SEARCHID&imgSize=medium&searchType=image&q=warhol+flowers>. When discussing the parameters, this points to the specification of the search API which reveals the full list of available parameters. However, this is an important step because most web APIs actually come with a specification that lists ways of interaction in the format of parameter/value pairs (Google Developers, 2020) as a matter of fact. For example, users can request grayscale images instead of coloured ones with the additional parameter of ‘imgColorType’ and the corresponding value as ‘grey’. From this simple parameter, we know that the image search service includes the categorisation of image colour. However, the specification can generate more discussion with the students, such as what parameters might be missing, and what we want to know more about regarding image selection and prioritisation.



**Figure 6.** The smaller version of *nag* that is used for teaching from the book titled ‘Aesthetic Programming’ (Soon and Cox, 2020).

```
1 let url = "https://www.googleapis.com/customsearch/v1?";
2 // register: https://developers.google.com/custom-search/json-api/v1/overview
3 let apikey = "INPUT YOUR OWN KEY";
4 //get the searchengine ID: https://cse.google.com/all (make sure image is on)
5 let engineID = "INPUT YOUR OWN".
```

```

6 let query = "warhol+flowers"; //search keywords
7 //check other parameters: https://tinyurl.com/googleapiCSE
8 let searchType = "image";
9 let imgSize = "medium";
10 let request; //full API
11
12 function setup() {
13   ...
14   fetchImage();
15 }
16
17 function fetchImage() {
18   request = url + "key=" + apikey + "&cx=" + engineID + "&imgSize=" + imgSize
19   + "&q=" + query + "&searchType=" + searchType;
20   console.log(request);
21   loadJSON(request, gotData); //this is the key syntax to make API request
22 }
23
24 function gotData(data) {
25   getImg = data.items[0].image.thumbnailLink;
26   console.log(getImg);
27 }

```

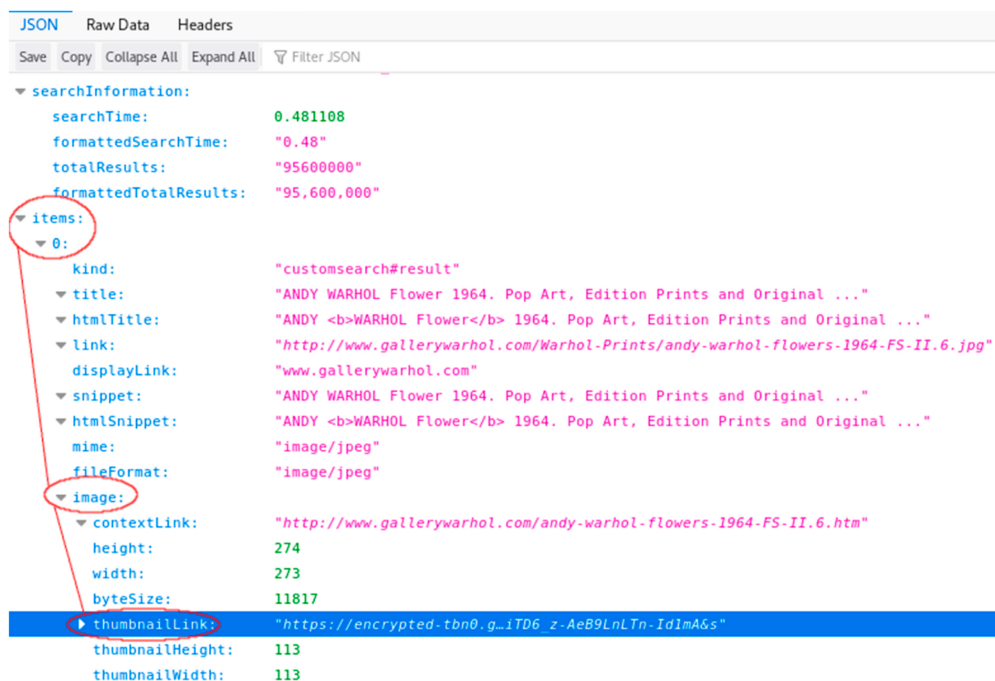
**Figure 7.** The snippet of code to illustrate what constitutes a web API.

### 3. Reading data files & Questioning data structures

The final step would be to (de)construct the data file format that is returned by Google. This data file format, in the form of a JSON file, provides an overview of the data structure and hierarchy (see Figure 6). Another important step in data processing is data parsing, which is to identify patterns or strings from the source that are needed. A returned JSON file from Google contains a lot of information, but this requires studying the data file in order to identify and extract the field (which is an image URL in this case) for display purposes in the program. But based on this file, we can also question why only 10 images are returned per request, and how images are prioritised to be seen and used among the millions of search results. Raising questions are important to challenge the assumptions and processes behind the data format. What are the ethical and affective consequences and implications of algorithms, such as the use of sorting algorithms in search engines? What are the parameters involved in sorting data? These are based on business' decision-making and personalised profiling which is highly commercialised. Noble has reminded us that Google Search results are normalised as 'believable and often presented as factual'. (Noble, 2018: 24–25) To follow the line of inquiry from Bellacase about the politics of care and maintenance work (Puig de la Bellacasa, 2017), perhaps we need to ask: Where is the line of responsibility and profit-making?

Although this is a specific case on Google Image Search API, the approach (or questions) can be adapted to use for investigating other platforms, such as Facebook, Tinder, Spotify, Twitter and many others but would probably need a specific attention to the platform infrastructure and to articulate the specific cultural and social relations. Through this end-to-end (de)construction from first getting API keys, then locating (un)available parameters and finally reading data files, students get hands-on experience with connecting APIs as technical objects to other processes, including accessing, tinkering, parsing and running code. Each step requires not only doing things technically but also thinking about and questioning the cultural and ethical relations and implications with regard to technical materials. In this section, we argue that critical technical practice is a *technical-discursive practice* as demonstrated above. Apart from engaging with

technical objects, It entangles the interpretations and questions the hierarchical processes that might not easily be made visible to others, and in which the entanglement of ‘matter and meaning cannot be severed’ (Barad and Tuin, 2012) (Figure 8).



**Figure 8.** Web API data structure I from the book titled ‘Aesthetic Programming’ (Soon and Cox, 2020).

### *What is technical in critical technical practice?*

Before we delve into the third technical object on Machine Learning, we may step back a little to rethink what is technical in critical technical practice. Philip Agre comes from a technical background with an expertise in artificial intelligence, but at the same time he recognises the need to engage with “nontechnical fields” such as social science, philosophy and liberal arts “beyond a popular level” (Agre, 1997: 145). He embraces the mix of science and humanities disciplines, as well as the intersection of technology and culture beyond technical formalisation. However, what does critical technical practice mean for people who have a humanistic background in which there is already a discursive training tradition? What is technical in critical technical practice for arts and humanities?

Arguably, knowledge about technical formalisation should not be limited to engineers and scientists, and there is also an increasing need in disciplines like the digital humanities to engage with digital tools and methods. According to Soon and Cox, “[o]n the one hand, we have a technical tradition that looks for precision and, on the other, there is a critical tradition that looks for ambiguity of meaning. It makes little sense to deny either approach” (Soon and Cox, 2020: 243–244). Soon and Cox do not address the use of technology as just tools, but stress the need to engage with the meaning of acquiring technical knowledge. As a physicist, Barad does not consider science and humanities to be in separated domains, but inherently entangled through their differences (Barad, 2012). The challenge then is how we can learn from one another and break the hierarchy between disciplines. Therefore, learning to code (such as creating a dataset frame with Python or querying an API as mentioned above, as well as tinkering machine learning models in the next section), as a form of material engagement, may also be a way to understand how machines operate in a

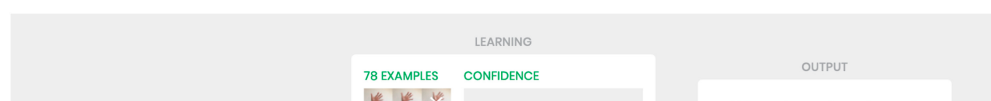
precise and concrete way that translates human interpretation into machine instructions, and to reflect on the forms of abstraction that are taking place.

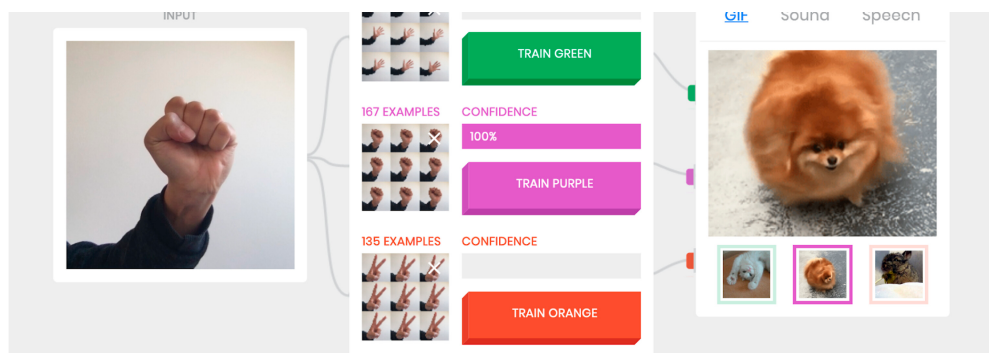
### *Machine learning*

Apart from the technical objects of datasets and APIs, that deal with rules and specification, machine learning builds upon the aforementioned data structures and further deals with three key data processes: input, models and output. Machine Learning is broadly defined as a collection of models, statistical methods and operational algorithms that are used to analyse data, and is related to cultural artefacts and practices, such as voice assistants, facial recognition and various recommendation engines that are deployed in apps and media services. Similar to APIs, machine learning is more than a technical object that depicts power dynamics as discussed in numerous scholarly works (Amoore, 2020; Azar et al., 2021; Benjamin, 2019; Burrell, 2016). It is an urgent topic in contemporary curriculums<sup>8</sup> to understand how machine learning might open up new possibilities to improve human life and ways of being, but also how it affects the way we see and experience the world, and what that means in relation to contemporary digital culture. The challenge is, again, how to introduce this complex technical-discursive object in a way that provides scope for reflections on the social structures and cultural phenomena that are entangled with the data processes, and to question existing technological infrastructure.

### **Understanding data processes**

Instead of introducing many technical terms of machine learning, an experimental AI project Teachable Machine<sup>9</sup> (version 1) is used as an interactive learning tool to explore the bodily relationship with data processes (see Figure 9). This specific tool comes with a clear and structured interface and playful outcomes. Unlike other machine learning tools that require separate software installation, Teachable Machine is a web application that runs on a web browser, utilising one's gestures and behaviours to provide the training dataset as the data input source. From the interface, one can clearly see how many frames have been used to train the model and it provides a maximum of three training classifiers to capture three distinct gestural input types. Furthermore, the training processes have also been mapped to multi-sensorial outputs as prediction outcomes, such as animated GIFs, sound clips and speech (audio files). In other words, the tool produces machine learning models via feeding input data and producing various outputs. With the playful interface, the classroom immediately turns into an active playground in which the students are required to move and explore their own bodily gestures, and they can see different animated images and sounds when they change their behaviours via the bodily interaction. Drawing upon critical studies of technoscience, Ratto and Hertz state that critical making is about material and conceptual exploration through the making of artefacts (Ratto and Hertz, 2019), and in this case, how the process of tinkering/making can facilitate the understanding of machine learning. But critical making is more than tinkering. It is also about drawing attention to the interwoven social- and technical aspects of the world by exploring how data is used, trained and predicted, and to further reflect on why things are done in a particular way by 'denaturalizing standard assumptions, values, and norms in order to reflect on the position and role of specific technologies within society.' (Ratto and Hertz, 2019: 22)





**Figure 9.** The Teachable Machine (Version 1) interface.

We asked the students to observe the predictive results that were shown as various outputs, and to test the boundaries of recognition or classification problems. The students were asked to examine what can, and cannot, be recognised under different conditions, such as variable lighting and distance, and the varying amount of trained data. In this way, tinkering with the tool is also about recognising the limitations and boundaries, and to further create a discursive space for questioning the technological processes.

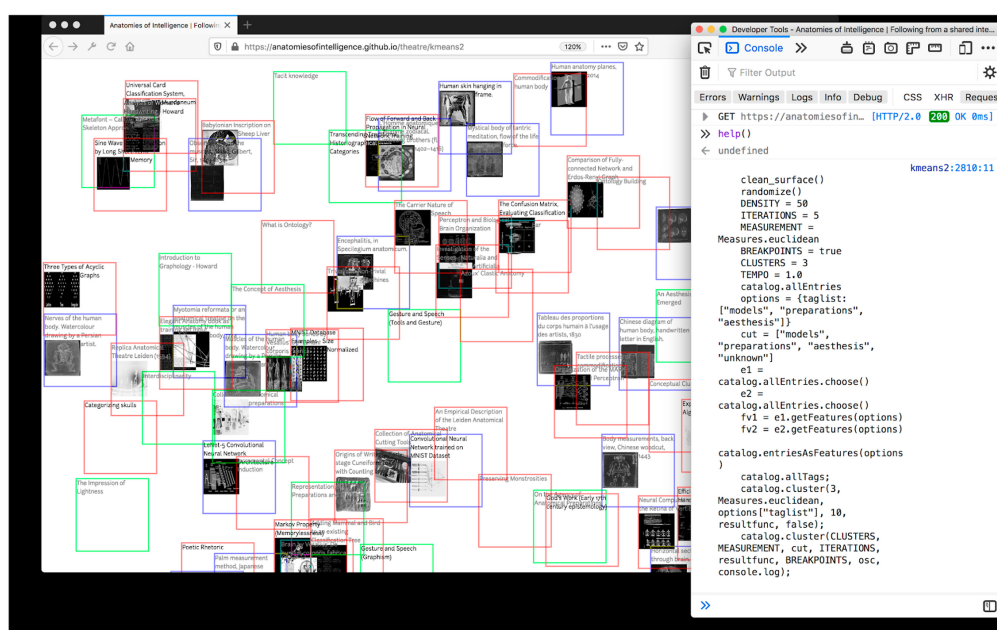
## Learning algorithmic modelling via critical art

Instead of simply introducing common types of machine learning models mathematically, such as supervised and unsupervised learning, the pedagogical approach, similar to how we introduced the artwork *nag* in the earlier API section, is to introduce art practice as forms of learning and knowing (Borgdorff, 2010; Sullivan, 1993; Pritchard and Prophet, 2015) in order to open up ways of working with machine learning otherwise. To highlight the scale of datasets, Nicolas Malevé's artwork entitled *12 Hours of ImageNet* (Malevé, 2019a) shows the labelling work that was completed by over 25,000 workers from a crowdsourcing platform called Amazon Mechanical Turk, addressing the labour conditions that are required to annotate and categorise in a short period of time, over 14 million photographs in the ImageNet dataset collected over 2 years. Unlike the common way of visualising data as graphs and charts, the artwork is a durational piece with a coded script to cycle through all the images of the ImageNet dataset at a speed of 90 milliseconds per image, but the screen pauses at random points to enable the observation of some of the images and their categorisation. In this way, some of the key technical concepts of supervised learning can be introduced at the same time, such as training datasets, labelling and discrete classification connected to a concrete application and research object, which in this case is the ImageNet as a large visual database, and further points to some of the matters of concern regarding image sources, labour conditions, labelling and categorisation (Malevé, 2019b).

Similarly, we have introduced an artwork entitled *Anatomies of Intelligence* by Joana Chicau and Jonathan Reus on unsupervised learning. Unlike supervised learning, which requires labelled data, the key concepts of unsupervised learning include clustering to find similar attributes and metrics that segregate groups. Through a collection of a much smaller image dataset on anatomy, the artwork explores new ways of putting images in clusters, and hence with a new perspective on experiencing the subject (anatomy) (see Figure 10). This new way of studying anatomy is explored through unsupervised learning, turning image features into data points, which makes it possible to measure the similarity between images and to generate various clusters. Through statistical analysis, the artists seek to 'make connections between the formats and collections of anatomical knowledge and investigation into the anatomy of computational learning and prediction processes, datasets and machine learning models' (Chicau and Reus, 2018). In other words, the



artists consider machine learning to be both an object and a subject of study, employing an explorative approach to seeking new forms of unknowable seeing.



**Figure 10.** Anatomies of Intelligence (2018–) by Joana Chicau and Jonathan Reus from the book titled ‘Aesthetic Programming’ (Soon and Cox, 2020). Courtesy of the artists.

The two artworks that have been discussed can be considered as critical art. Both artists position themselves as artists and researchers, breaking the hierarchy from the theory-practice dichotomy, and taking the subject of machine learning into critical inquiry via exploration and experimentation. Such non-normative ways of making, which are different from solutionist and industry/market-driven approaches, promote deeper engagement and critical reflection on the sources and materials by using creative ways to explore the subject matter through, and with, technology. In the fields of Software Studies and Digital Culture, one could also argue that art plays a major role in the formation of the disciplines (Andersen and Pold, 2018; Fuller, 2003; Goriunova, 2013; Goriunova and Shulgin, 2004), expanding the notion of art from confined gallery spaces to academic discourses. Such critical art practice exposes the underlying black box of machine learning processes and conditions by delving into the technical making, shedding light on how one might work and think with technical objects/subjects.

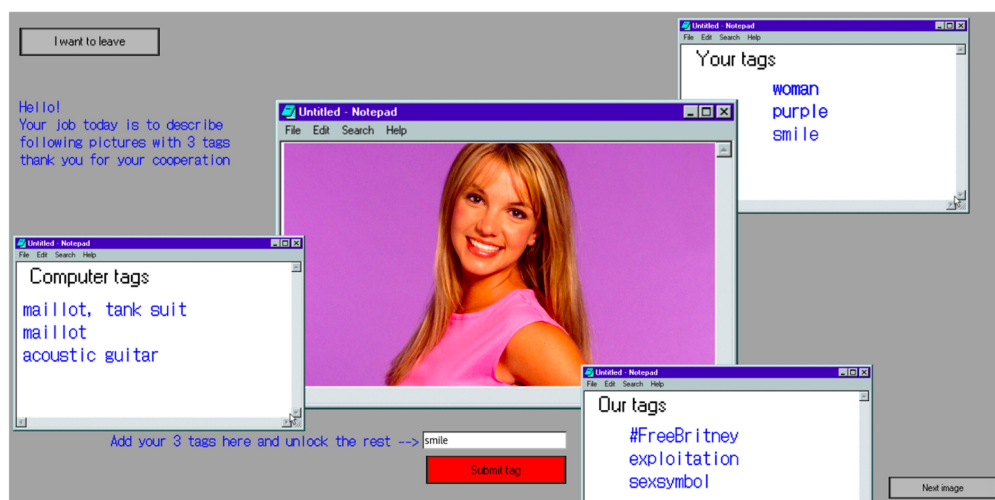
## Constructing critical artefactscal

After the playful experience with the Teachable machine tool and introducing some of the basic concepts of machine learning with critical arts, the last pedagogic approach is to work on something very hands-on with the introduction of the ml5.js, which is a machine learning JavaScript library designed for beginners and to make machine learning accessible. By using customised sample code, students are able to run the program (similar to the use of the Teachable Machine tool) but at the same time to read at the source code level in order to learn how the machine learning works. For example, they will look into the commands and syntaxes required for the three key processes: input, models and output. Posing questions is an important technique to create the discursive space in the classroom, for example: What are the inputs needed if this is not gesture recognition but another type of prediction, such as text? What are the models required to learn the writing

style of someone? What would be needed to output predictive text? What are the basic parameters to adjust the format and experience of prediction?

The M15.js library provides a comprehensive sample code and technical schemas that cover machine learning in relation to image, sound and text. As part of the Aesthetic Programming course, students will find it easier to read the syntaxes and commands as they have already been introduced to the basic programming principles at their earlier stages. Therefore, having the skills to program also allows students to express their ideas and be attentive to their decision making via the construction of artefacts. It requires both technical and discursive practice skills to contextualise the matters of concern and put ideas in a concrete form (a program). One of the groups explores ‘Tagging Bias’ in their final project (see Figure 11) by using the trained model from ImageNet to rethink the labour conditions and bias of tagging in machine learning. Users can interact with the software program that shows a selected image with three tags, displaying the discrepancy between three tagging agents: First the users (‘Your tags’ in Figure 11), then the authors of the program (‘Our tags’ in Figure 11), and finally the trained machine learning model (‘Computer tags’ in Figure 11). It is argued that constructing artefacts by engaging with technical materials can produce critical inquiry and reflexive thinking, as the students put it,

“For this whole process we have been very aware of our biases, and the decision to label the pictures ourselves underlines the lack of objectivity. In this process it has been important for us to be aware and transparent of our own role in the program. This process was not as easy as we had first thought, but this also accentuates the importance and responsibility when labelling pictures.” (Bolt et al., 2021: 7)



**Figure 11.** *Tagging Bias* (2021) by Nina Isis Kinch Bolton, Sofie Lundby Andersen, Simon Feusi, Sofie Fürsterling Mønster and Mathilde Borregaard Gajhede.

The above example clearly shows that decisions need to be practically and conceptually considered via constructing critical artefacts. Beyond art/technical making, we can also see how art-based learning might incorporate critical technical practice as a pedagogical approach, emphasising critical inquiry, reflective processes and conceptual ideas rather than end products in order to open up a different way of engaging with cultural and societal issues. We argue that Agre’s critical technical practice can be expanded from a technical research domain to classroom settings for arts and humanities. In other words, doing or discussing critical technical practice is also a means of conceptual exploration that seeks to ignite thinking, pose questions and expose social-technical conditions rather than solving particular well-defined problems. To interpret Agre’s

words, engaging with technical materials like the mix of artworks, vocabulary, tools, schema and representation “provides a way of talking about a very wide range of phenomena in the world” (1997: 143) and reflecting on how machine learning models the world.

To conclude this section, we have discussed the pedagogical approach we used in examining machine learning as a technical-discursive object, from playful and bodily experience to critical art and technical construction with the aim of rethinking what critical technical practice is in arts and humanities. In addition to the attention we have given to criticality throughout the whole article, we also argue that engaging with technical materials necessitates practical understanding and knowledge of socio-technical systems. There are many ways to engage with technical materials, but we have chosen to do so via (de)constructing machines as part of doing critical practice with the aim of further unsettling the abstract interpretation and concrete formalisation in different disciplines. We hope to open up the space of ‘Critical’ ‘Technical’ ‘Practice’, unfolding aspects of both formality and functionality as well as questioning and understanding technology at a discursive and critical level.

## Conclusion

Although we distinguish above between different technical objects (dataset, API, machine learning) for analytical and reflective purposes, it is important to highlight the interdependence, or *technical coherence*, of the former objects. As an example, the Teachable Machine is not a standalone object in this thread. The project relies on Google financially, and as such it follows its privacy policies and terms of agreement. Beyond the organisational reliance, the same project is made technically possible through tensorflow.js modelling, a common library for data science and machine learning. AI-based image recognition and image generation is, in turn, made possible through several iterations of models built on Imagenet (Deng et al., 2009), a canonical image dataset which inherits many of its categories from Wordnet (Fellbaum, 1998; Miller, 1998). Moreover, communication between all these technical objects datasets and models, interfaces and queries, etc – usually happens via APIs. We want to insist not only on the technical inter-reliance between these objects (Helmond, 2015; Van Dijck, 2020), but particularly on the formalisation of specific types of knowledge that are solidified in these iterations.

An applied Critical Technical Practice is aware of how these iterations in the design and implementation of technical objects (Agre, 1994) structure novel forms of knowledge. Agre identifies, for example, how Cartesian accounts of cognitive rationality were able to explain mental and physical phenomena by reconfiguring (‘rendered mechanical’) knowledge problemata as method (Agre et al., 1997: 136). These Cartesian *grammars*, or technical formalisations for epistemological approaches, are techniques that act as both ‘a method for designing artefacts and a thematics for narrating its operation’ (Agre, 1994: 141). Purely instrumental and methodological approaches to technical objects and technical schemata (eg mathematical representations) follow these grammar to some degree. That is, technical design and implementation not only put into action changes in the social world – *for what purpose is a certain database made?* – but also guide the possibility of action – *which kind of discursive subjects are database-compatible?* – within their own narratives.

Our (de)constructive approach aims to consider the former complexity in a classroom setting: it follows the technical elements for the knowledge of technical objects, but in doing so puts into question the grammars associated with them. In this sense, a *critical* technical approach cannot be identified merely as ‘data literacy’, but as the identification of existing and developing grammars (or embedded rules), power



structures and practices that come with such knowledge, where ‘action and datafication are thus designed to be co-constitutive’ (Gray et al., 2018: 5).<sup>10</sup> Moreover, our take on Critical Technical Practice integrates feminist considerations into the study of technical entities, as it is detached from modern notions of fact, properties of objects, simple representations, instrumentality and non-situated claims, and towards matters of concern, material-discursive entanglements, intervention, and situated practices. Following critical practice and critical making traditions, our work develops a pedagogical approach to understanding the entanglement between digital objects, bodies, society and knowledge, by focusing on reflexive dissections, situated agency, playful tinkering and observations *beyond – yet within* – their technical elements.

## Acknowledgements

Both authors contributed equally to this work. An earlier version of this article was presented at the online workshop on ‘Critical Interface Analysis’ in 2020 (10-12 August, Aarhus University). We would like to thank all the reviewers who have provided very constructive and detailed comments.

## ORCID iDs

Winnie Soon      <https://orcid.org/0000-0002-0790-4737>

Pablo R Velasco      <https://orcid.org/0000-0003-2140-3733>

## Notes

- 1 Our use of the term does not relate directly to the more traditional uses of ‘deconstruction’ in literature. While we share the general understanding of deconstruction as an analytical tool to develop a critique of hardened categories (e.g. the early heideggerian use of *destruktion*), our use of the term does not refer to linguistic elements as developed by Derrida’s *Of Grammatology*. Even if there are certain points in common with the traditional conception of deconstruction, such as the distributive nature of meaning, we focus on (*de*)*construction* to stress the materiality of the technical objects we engage with, and understand this term mainly as a methodological tool to scrutinise them through disassembling and reassembling. From here on, we add parentheses at the beginning of the word to signal this particular use.
- 2 AP and DS are courses within the department called Digital Design and Information Studies at Aarhus University, Denmark. AP is a practice-oriented course on the fundamentals of programming, exploring the relationship between code and cultural and/or aesthetic issues while DS deals with the question of what can be understood as ‘data’ regarding its collection, analysis, uses and methodological approaches.
- 3 While the most common visual representation of datasets are tables, they also may commonly be stored as a string (e.g. a json file), keeping their machine-readability properties and general ‘dataset’ logic.
- 4 This section about the API technical object is adapted from the chapter titled Que(e)ry Data in the book *Aesthetic Programming: A Handbook of Software Studies* (Soon and Cox, 2020). See all the steps, source code and exercises here: <https://aesthetic-programming.net/pages/8-queuey-data.html>
- 5 For example, artist and educator Lauren Lee McCarthy developed the course ‘Critical APIs’ as part of the Interactive Telecommunications Programs (2015). Ruben van de Ven, Max Dovey, Manetta Berends conducted a workshop on ‘API-art & situated fictions’ for the Saint Joost Academy in 2017. Winnie Soon and Geoff Cox included the chapter ‘Data qu(e)ry’ about search APIs in their book *Aesthetic Programming* (2020), and Soon is also collaborating with Helen Pritchard on conducting a workshop on ‘Queer API’ for the Piet Zwart Institute in 2021. In the same year, Cristina Cochior, Manetta Berends, Michael Murtaugh and Steve Rushton led the Experimental Publishing Master module (also from Piet Zwart Institute) with a focus on a more critical approach to data distribution via API, see the outcome of the course: X PUB special Issue 16 on vernacular language processing: <https://hub.xpub.nl/soupboat/si16/about/>

- 6 Techno Mourning was the title of a blog in 2015 for documenting art projects that relied specifically on Google Earth API in order to operate, because Google had announced the deprecation of the Google Earth-plugin or API in December 2014. See the project here: <http://www.250miles.net/techno-mourning/>
- 7 See the context and background of the artwork here: <https://net.art-generator.com/>
- 8 This section about the Machine Learning technical object is adapted from the chapter titled Machine unlearning in the book *Aesthetic Programming: A Handbook of Software Studies* (Soon and Cox, 2020). See some of the steps, source code and exercises here: <https://aesthetic-programming.net/pages/10-machine-unlearning.html>
- 9 We acknowledge there are already many machine learning tools that have been used in classroom teaching, for example the free and open source software Wekinator developed by Rebecca Fiebrank and the free game while True: learn() developed by Valve Corporation. We have used the web based Google Teachable machine version 1 because we only have 2 class sessions to focus on machine learning and the students are still learning the basics of web programming. As such we need to take into consideration the selection of tools with the entire course structure. See here: <https://teachablemachine.withgoogle.com/v1/>
- 10 Gray et al. (2018), refer to this approach as ‘data infrastructure literacy’.

## References

- Agre PE (1994) Surveillance and capture: Two models of privacy. *The Information Society* 10(2): 101–127. Crossref
- Agre PE (1997) Toward a critical technical practice: Lessons learned in trying to reform AI. In: Bowker GC, Star SL, Turner W, et al. (eds), *Social Science, Technical Systems and Cooperative Work*. Erlbaum.
- Amoore L (2020) *Cloud Ethics: Algorithms and the Attributes of Ourselves and Others*. Durham: Duke University Press.
- Andersen CU, Pold S (2018) *The Metainterface: The Art of Platforms, Cities, and Clouds*. London, England: The MIT Press.
- Asano S, Yashiro T, Sakamura K (2016) A proxy framework for API interoperability in the Internet of Things. In: 2016 IEEE 5th Global Conference on Consumer Electronics, Kyoto, Japan, October 2016. IEEE, pp. 1–2. Crossref
- Azar M, Cox G, Impett L (2021) Introduction: ways of machine seeing. *AI & SOCIETY* 36: 1093–1104. Crossref
- Barad K (2007) *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Duke University Press.
- Barad K, Tuin RD (2012) I van der “Matter feels, converses, suffers, desires, yearns and remembers” Interview with Karen Barad. In: *New Materialism: Interviews & Cartographies*. Available at: <http://hdl.handle.net/2027/spo.11515701.0001.001>
- Beer D (2018) *The Data Gaze: Capitalism, Power and Perception*. 1 edition. Thousand Oaks, CA: SAGE Publications Ltd.
- Benjamin R (2019) *Race after Technology: Abolitionist Tools for the New Jim Code*. Medford, MA: Polity.
- Bolt NIK, Andersen SL, Feusi S, et al. (2021) *Tagging Bias*. Available at: [https://gitlab.com/Sofiefm/aesthetic-programming/-/blob/master/finalProject/FINALPROJECT\\_AP\\_\\_1\\_.pdf](https://gitlab.com/Sofiefm/aesthetic-programming/-/blob/master/finalProject/FINALPROJECT_AP__1_.pdf)
- Borgdorff H (2010) The production of knowledge in artistic research. In: Biggs M, Karlsson H, Riksbankens jubileumsfond (eds), *The Routledge Companion to Research in the Arts*. 1st ed. New York: Routledge, pp. 44–63.
- Bucher T (2013) *Objects of Intense Feeling: The Case of the Twitter API*. Computational Culture.
- Bucher T (2018) *If...Then: Algorithmic Power and Politics*. New York: Oxford University Press.
- Burkhardt M, Helmond A, Seitz T, et al. (2020) *The Evolution of Facebook’s Graph Api*. AoIR Selected Papers of Internet Research. Crossref
- Burrell J (2016) How the machine ‘thinks’: Understanding opacity in machine learning algorithms. *Big Data & Society*

3(1): 205395171562251. Crossref

Cheney-Lippold J (2017) *We Are Data: Algorithms and the Making of Our Digital Selves*. New York: NYU Press..

Chicau J, Reus J (2018) *Anatomies of Intelligence*. Available at: (accessed on 13 September 2021) <https://anatomiesofintelligence.github.io/>

Cramer F, Fuller M (2008) Interface. In: *Software Studies: A Lexicon*. Cambridge, MA: MIT Press.

Dalton C, Thatcher J (2014) What does a critical data studies looks like, and why do we care? In: *Society & Space*. Available at: <http://societyandspace.org/2014/05/12/what-does-a-critical-data-studies-look-like-and-why-do-we-care-craig-dalton-and-jim-thatcher/> (accessed 24 December 2018).

Deng J, Dong W, Socher R, et al. (2009) ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition 2009*: 248–255. Crossref

D'Ignazio C, Klein LF (2020) Data Feminism. In: *Strong Ideas Series*. Cambridge, MA: The MIT Press.

Dijck J van (2014) Datafication, dataism and dataveillance: Big Data between scientific paradigm and ideology. *Surveillance & Society* 12(2): 197–208. Crossref.

Dourish P (2014) *The Shifting Materialities of Database Technology*. *Computational Culture*. No SQL. Available at: <http://computationalculture.net/article/no-sql-the-shifting-materialities-of-database-technology> (accessed 28 September 2017).

Dourish P, Finlay J, Sengers P, et al. (2004) Reflective HCI. In: Extended Abstracts of the 2004 Conference on Human Factors and Computing Systems - CHI '04, Vienna, Austria, 2004, ACM Press, p. 1727. Crossref

Drucker J (2014) *Graphesis: Visual Forms of Knowledge Production*. Cambridge, MA: Harvard University Press.

Fellbaum C (1998) A Semantic Network of English: The Mother of All WordNets. In: Vossen P (ed), *A Multilingual Database with Lexical Semantic Networks*. Dordrecht: Springer, pp. 137–148. Crossref

Finn E (2017) *What Algorithms Want: Imagination in the Age of Computing*. Cambridge, MA: MIT Press.

Foucault M (1971) Orders of discourse. In: *Sage Publications Sage CA*. CA: Thousand Oaks, 10, pp. 7–30.

Foucault M (1978) *Discipline and Punish: The Birth of the Prison*. Knopf Doubleday Publishing Group: Vintage. Available at: <https://books.google.co.uk/books?id=6rfPOH5TSmYC>

Fuller M (2003) *Behind the Blip: Essays on the Culture of Software*. Brooklyn, NY: Autonomedia.

Gitelman L (2013) *Raw Data Is an Oxymoron*. MIT Press.

Gollihue K, Browning A (1988) Towards an Intersectional Feminist Critical Making. *The Critical Makers Reader:(Un) Learning Technology* 14: 227.

Google Developers (2020) *cse.list | Custom Search JSON API*. Available at: <https://developers.google.com/custom-search/v1/reference/rest/v1/cse/list> (accessed 13 September 2021).

Goriunova O (2013) Art Platforms and Cultural Production on the Internet. 1. iss. in paperback. In: *Routledge Research in Cultural and Media Studies* 35. New York: Routledge.

Goriunova O (2019) The Digital Subject: People as Data as Persons. *Theory, Culture & Society* 36(6): 125–145. Crossref

Goriunova O, Shulgin A (eds), (2004) *Digital Aesthetics Research Centre*. University of Aarhus.

Gray J, Gerlitz C, Bounegru L (2018) Data infrastructure literacy. *Big Data & Society* 5(2): 205395171878631. Crossref

Gurses S, van Hoboken JVJ (2017) *Privacy after the Agile Turn*. preprint, 2 May. SocArXiv. Crossref

Hacking I (1986) Making up people. na.

Haigh T (2002) Software in the 1960s as concept, service, and product. *IEEE Annals of the History of Computing* 24(1): 5–13. Crossref

- Haraway D (1988) Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective. *Feminist Studies* 14(3): 575. Crossref
- Harwood G (2019). In: Bogers L, Chiappini L (eds), *Teaching Critical Technical PracticeThe Critical Makers Reader: (Un)Learning Technology. INC Reader 12*. Amsterdam: Institute of Network Cultures.
- Helmond A (2015) The Platformization of the Web: Making Web Data Platform Ready. *Social Media + Society* 1(2): 2056305115603080. Crossref
- Holmes NG, Wieman CE, Bonn DA (2015) Teaching critical thinking. *Proceedings of the National Academy of Sciences* 112(36): 11199–11204.
- Hui Y (2012) What is a Digital Object? *Metaphilosophy* 43(4): 380–395. Crossref
- Iliadis A, Russo F (2016) Critical data studies: An introduction. *Big Data & Society* 3(2): 205395171667423. Crossref
- Johnston J (2008) The Allure of Machinic Life: Cybernetics. In: *Artificial Life, and the New AI*. MIT Press.
- Katzenbach C (2021) “AI will fix this” - The Technical, Discursive, and Political Turn to AI in Governing Communication. *Big Data & Society* 8(2): 205395172110461. Crossref
- Kinsella EA (2007) Embodied Reflection and the Epistemology of Reflective Practice. *Journal of Philosophy of Education* 41(3): 395–409. Crossref
- Kitchin R (2014) *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*. SAGE.
- Kitchin R, Lauriault TP (2014) *Towards Critical Data Studies: Charting and Unpacking Data Assemblages and Their Work*, 19.
- Latour B (2004) Why Has Critique Run out of Steam? From Matters of Fact to Matters of Concern. *Critical Inquiry* 30(2): 225–248. Crossref
- Latour B (2007) *Reassembling the Social: An Introduction to Actor-Network-Theory*. OUP Oxford.
- Latour B (2017) Gaia or Knowledge without Spheres. In: Schaffer S, Tresch J, Gagliardi P (eds). *Aesthetics of Universal Knowledge*. Cham: Springer International Publishing, 169–201. Crossref.
- Malev  N (2019a) *12 Hours of ImageNet*. Available at: <https://www.youtube.com/watch?v=PC60JL-IMzA>
- Malev  N (2019b) *An Introduction to Image Datasets*. Available at: <https://unthinking.photography/articles/an-introduction-to-image-datasets> (accessed on 8 October 2020).
- Manovich L (2001) *The Language of New Media*. MIT Press.
- Marres N, Weltevrede E (2013) Scraping the Social? *Journal of Cultural Economy* 6(3): 313–335.
- Mauri M, Elli T, Caviglia G, et al. (2017) RAWGraphs: A Visualisation Platform to Create Open Outputs. In: Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter, New York, NY, USA, 18 September 2017, pp. 1–5. CHIItaly ’17. Association for Computing Machinery. Crossref.
- McInerny G (2018) Visualizing data. In: Lury C, Fensham R, Heller-Nicholas A, et al. (eds), *Routledge Handbook of Interdisciplinary Research Methods*. 1st ed. Routledge, pp. 133–141. Crossref
- Mejias UA, Couldry N (2019) Datafication. *Internet Policy Review* 8(4): 1. Crossref
- Miller GA (1998) *WordNet: An Electronic Lexical Database*. MIT Press.
- Noble SU (2018) How Search Engines Reinforce Racism. In: *Algorithms of Oppression*. New York: New York University Press.
- Passi S, Jackson S (2017) Data Vision: Learning to See Through Algorithmic Abstraction. In: Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, Portland Oregon USA, 25 February 2017, pp. 2436–2447. ACM. Crossref

- Pritchard H, Prophet J (2015) *Diffractional Art Practices: Computation and the Messy Entanglements between Mainstream Contemporary Art, and New Media Art\**. Artnodes, 15. Crossref
- Puig de la Bellacasa M (2017) *Matters of Care: Speculative Ethics in More than Human Worlds*. Posthumanities 41. Minneapolis: University of Minnesota Press.
- Raetzsch C, Pereira G, Vestergaard LS, et al. (2019) Weaving seams with data: Conceptualizing City APIs as elements of infrastructures. *Big Data & Society* 6(1): 205395171982761. Crossref
- Ratto M (2011) Critical Making: Conceptual and Material Studies in Technology and Social Life. *The Information Society* 27(4): 252–260. Crossref
- Ratto M, Hertz G (2019) Critical Making and Interdisciplinary Learning: Making as a Bridge between Art, Science, Engineering and Social Interventions. In: Bogers L, Chiappini L (eds), *The Critical Makers Reader: (Un)Learning Technology*. Amsterdam: the Institute of Network Cultures, pp. 17–28.
- Ratto M, Hockema S (2009) Flwr Pwr: Tending the Walled Garden. In: Dekker A, Wolfsberger A (eds), *Walled Garden*. Amsterdam: Virtueel Platform, pp. 51–62.
- Ratto M, Wylie SA, Jalbert K (2014) Introduction to the Special Forum on Critical Making as Research Program. *The Information Society* 30(2): 85–95. Crossref
- Rogers R (2019) *Doing Digital Methods*. Thousand Oaks, CA: SAGE Publications Ltd.
- Ruppert E (2013) Not Just Another Database: The Transactions that Enact Young Offenders. *Computational Culture*: 1–13.
- Ruppert E, Law J, Savage M (2013) Reassembling social science methods: The challenge of digital devices. *Theory, Culture & Society* 30(4): 22–46.
- Seaver N (2019) Knowing Algorithms. In: Vertesi J, Ribes D (eds), *DigitalSTS: A Field Guide for Science & Technology Studies*. Princeton & Oxford. Princeton University Press, pp. 412–422.
- Siegel H (2017) *Cultivating Reason*. Oxford University Press. Available at: <https://www.oxfordscholarship.com/view/10.1093/oso/9780190682675.001.0001/oso-9780190682675-chapter-1> (accessed on 9 December 2019).
- Snodgrass E, Soon W (2019) API practices and paradigms: Exploring the protocological parameters of APIs as key facilitators of sociotechnical forms of exchange. *First Monday*. Crossref
- Sollfrank C, Soon W (2021) *Fix My Code*. Berlin: EECLECTIC. Available at: <https://eeclectic.de/produkt/fix-my-code/>
- Soon W (2016) *Executing Liveness: An Examination of the Live Dimension of Code Inter-actions in Software (Art) Practice*. Aarhus: Aarhus University. Available at: [http://siusoon.net/home/me/doc/soon\\_PhD\\_FINAL.pdf](http://siusoon.net/home/me/doc/soon_PhD_FINAL.pdf)
- Soon W, Cox G (2020) *Aesthetic Programming: A Handbook of Software Studies*. London: Open Humanities Press.
- Soon W, Pritchard H (2021) *Queer Motto API*. Available at: (accessed on 1 September 2021) <https://gitlab.com/siusoon/queer-motto-api>
- Sullivan D (2016) *Google Now Handles at Least 2 Trillion Searches Per Year*. Available at <https://searchengineland.dev/google-now-handles-2-999-trillion-searches-per-year-250247> (accessed on 13 September 2021).
- Sullivan G (1993) Art-Based Art Education: Learning That is Meaningful, Authentic, Critical and Pluralist. *Studies in Art Education* 35(1): 5–21. Crossref
- Uprichard E (2013) *Focus: Big Data, Little Questions?* in: *Discover Society*. Available at: <https://archive.discoversociety.org/2013/10/01/focus-big-data-little-questions/> (accessed on 12 April 2022).
- Van Dijck J (2020) Seeing the forest for the trees: Visualizing platformization and its governance. In: *New Media & Society*. SAGE Publications, 23, pp. 2801–2819. Crossref.
- Yegge S (2011) *Stevey's Google Platforms Rant*. Available at: <https://gist.github.com/chitchcock/1281611> (accessed on 1

September 2021).