International Journal of Communication 10(2016), 4972–4990 1932–8036/20160005

When the Algorithm Itself Is a Racist: Diagnosing Ethical Harm in the Basic Components of Software

CHRISTIAN SANDVIG[1]
University of Michigan, USA

KEVIN HAMILTON
KARRIE KARAHALIOS
CEDRIC LANGBORT
University of Illinois at Urbana-Champaign, USA

Computer algorithms organize and select information across a wide range of applications and industries, from search results to social media. Abuses of power by Internet platforms have led to calls for algorithm transparency and regulation. Algorithms have a particularly problematic history of processing information about race. Yet some analysts have warned that foundational computer algorithms are not useful subjects for ethical or normative analysis due to complexity, secrecy, technical character, or generality. We respond by investigating what it is an analyst needs to know to determine whether the algorithm in a computer system is improper, unethical, or illegal in itself. We argue that an "algorithmic ethics" can analyze a particular published algorithm. We explain the importance of developing a practical algorithmic ethics that addresses virtues, consequences, and norms: We increasingly delegate authority to algorithms, and they are fast becoming obscure but important elements of social structure.

Keywords: applied ethics, information and communication technologies (ICT), science and technology studies (STS), Internet studies, algorithms

Christian Sandvig: csandvig@umich.edu
Kevin Hamilton: kham@illinois.edu
Karrie Karahalios: kkarahal@cs.uiuc.edu
Cedric Langbort: langbort@illinois.edu
Date submitted: 2016–08–10

[1] This article began as a response to Daniel Neyland at the NYU symposium "Governing Algorithms." The authors would like to thank the organizers, Philip Howard, Mike Ananny, and the anonymous peer reviewers. An earlier version of this article was presented as a conference paper to the 65th annual meeting of the International Communication Association. This work was funded in part by the Interdisciplinary Innovation Initiative at the University of Illinois. The authors gratefully acknowledge the feedback from the organizers of the daylong workshop "Algorithms, Automation and Politics," organized by the European Research Council–funded Computational Propaganda project of the Oxford Internet Institute and held as a preconference to the International Communication Association Meeting in Fukuoka, Japan, in June 2016. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the European Research Council.

Hewlett-Packard (HP) suffered a serious public relations crisis when it was revealed that its implementation of what was probably a bottom-up feature-based face localization algorithm (Yang, Kriegman, & Ahuja, 2002) did not detect Black people as having a face (Simon, 2009). Cameras on new HP computers did not track the faces of Black people in some common lighting conditions. In an amusing YouTube video with millions of views, Wanda Zamen (who is White) and Desi Cryer (who is Black) demonstrate that the HP camera eerily tracks Zamen's face while ignoring Cryer, leading Cryer to exclaim jokingly, "Hewlett-Packard computers are racist" (Zamen, 2009).

Dyer (1997), a historian of cinema, famously came to a similar conclusion about photographic technology. In his classic studies of photographic and cinematic representation of human skin, he explained that, as early photographers turned to portraiture in the 1840s, "experiment[s] with, for instance, the chemistry of photographic stock, aperture size, length of development, and artificial light all proceeded on the assumption that what had to be got right was the look of the white face" (p. 90). Technological investments followed value: to put it starkly, Black people were not considered to be worth photographing. At the dawn of the camera the representation of White skin was seen as very difficult due to its tendency to wash out or shade to unrealistic red tones, but with effort these difficulties were solved; White became the norm; and photographing non-White people has been considered the problem or the exception ever since. As late as the 1970s, 3M Corporation developed a television signal named "skin" (a pale orange color) used to grade the quality of videotape (Dyer, 1997, p. 94).

These two cases are obviously different—Zamen and Cryer demonstrated an embarrassing HP oversight that was quickly corrected with a software update. In contrast, Dyer unearthed evidence that decisions made in the early days of photography embedded racist assumptions in chemical formulations, apparatuses, and processes that led to 150 years (and counting) of difficulty for non-White photographic subjects. Yet in each instance, a technological system becomes the vehicle for embedded human social dynamics (racism), which it could then perpetuate. In both cases, the system produces effects that are clearly ethically worrisome. Considering intent or assigning human responsibility is fruitless in both cases. That is, neither the companies involved nor the individuals within them probably intended to be racist. If we identify specific inventors of photographic apparatuses from the 1840s, they would only be exemplars of the attitudes common at that time. The developers of the HP camera saw this behavior as an error and apologized. The lack of testing on Black skin in whatever HP department develops webcams does seem to be a notable mistake, however. Understanding the dynamics of both cases in detail requires some engagement with the technical innards of the system. Whether photography in the 1840s had algorithms is a question that will be addressed later in this article, yet Dyer manages his analysis and critique without resorting to the word or the idea of an algorithm.

Today, computer algorithms play a critical role in producing and curating our communications and shared culture (Ziewitz, 2015). They determine how our questions are answered (Introna & Nissenbaum, 2000); decide what is relevant for us to see (Gillespie, 2012); craft our personal and professional networks (Hamilton, Karahalios, Sandvig, & Eslami, 2014); suggest who we should date and what we should watch; and profile our behavior to determine what advertising we will receive (Bermejo, 2007). Research and writing is now blossoming among academics, journalists, and nonprofit organizations that criticizes (Gangadharan, 2014; Pasquale, 2015) and even reverse-engineers (Angwin, n.d.; Diakopoulos,

2014; Hannack et al., 2013) these algorithms in an attempt to see inside these new systems and understand their consequences (for an overview, see Sandvig, Hamilton, Karahalios, & Langbort, 2014). There is some precedent for administrative law regulations that forbid particular processes in algorithms: In the history of aviation, travel-booking systems have been forbidden to deploy sorting algorithms that display more profitable itineraries over shorter, cheaper itineraries. In sum, policy scholars are now arguing for the regulation of algorithms as a distinct source of harms (Grimmelmann, 2009).

When the Algorithm Itself Is a Problem

Despite the general enthusiasm for investigating algorithms, most research does not consider actual computer algorithms that are now in use. Valuable scholarship has focused on the implications of the concept of the algorithm, and this sometimes includes examples of algorithms (Totaro & Ninno, 2014). Algorithmic systems are widely discussed, but algorithms per se are often not considered. For the sake of philosophical argument, an algorithm is either specified hypothetically in general terms or it is ruled to be off-limits as inaccessible to the analysts (Floridi & Sanders, 2004). Some investigators have gone so far as to warn researchers away from considering algorithms in themselves, arguing that this elides the embeddedness of algorithms within a sociotechnical system (Neyland, 2016; Neyland & Mollers, 2016; Seaver, 2014). Other approaches to develop new algorithmic ethics have explicitly cautioned against reading code (Ananny, 2015). How much do researchers who study these systems actually need to know about real algorithms?

Although the algorithm has emerged as an important concept in the public mind (Sandvig, 2014; Striphas, 2015), it seems reasonable that scholars of algorithmic culture (a term coined by Galloway, 2006) might study the consequences of the addition of computing to these media and information systems without needing to know the specifics of process involved in a low-level component in a computer system. But we argue that there is important new knowledge to be gained by considering the operation of algorithms writ small—the technical details of the innards of particular computer code. To accomplish this analysis, we have assembled a multidisciplinary team of coauthors (a computer scientist, an applied mathematician, an artist, and a social scientist) to analyze the problems of assigning ethical responsibility to algorithms in general; then we review one case in detail: the choice of a computer vision algorithm in a hypothetical surveillance system.

Was the Algorithm Racist?

Did HP's digital camera have a racist algorithm? The manufacturer had a wide range of preexisting approaches to localization at its disposal (Yang et al., 2002)—*localization* being the term for the problem of detecting the presence and location of particular features (faces) in an image, in contrast to *recognition* (matching them to another face). HP could have chosen a bottom-up feature-based algorithm—likely the intellectual descendant of Chetverikov and Lerch (1992), who realized that human faces could be successfully detected by asking a computer to find a pattern of dark and light blobs indicating the eyes, cheekbones, and nose. Because this tactic depends on contrast between the skin and the whites of the eyes, dark skin presents a challenge.

Other algorithms for locating a face make different trade-offs. Detecting faces with a predefined palette of skin colors is an alternative approach (Graf, Chen, Petajan, & Cosatto, 1995). Using histogram intersection between a control skin and a target will be very sensitive to the definition of what counts as a skin color. Since these approaches require an explicit definition of skin and non-skin colors in advance, it logically follows they could certainly be racist. In contrast, edge detection methods might be much more dependent on the contrast between the body and the background. They could be sensitive to hair color and a contrasting background color, making skin color relatively unimportant. A third approach would be an algorithm for determining and displaying skin color based on machine learning. This might seem a desirable solution, but we can just as easily imagine such a process coming to objectionable conclusions on its own depending on the training data it was given.

Although each of these algorithmic approaches has consequences for efficiency and accuracy when evaluated in specific situations, we can also say that the choice of algorithm delegates the ability to define a face. Though one could argue that an ethical critique of algorithms should focus more on the design process of a system than on the resulting technology, we believe that the nature of algorithms invites a closer examination of that very distinction. To neglect the algorithm in the study of an information system's ethics overlooks the significance of algorithms as singularly important statements, ways of framing problems and assuming not only particular implementations but ways of dealing with undesirable results. Goffey (2008) has described algorithms as key examples of Foucault's mandate to interrogate and situate such statements. But what do computer scientists mean when they use the word *algorithm*?

Defining Algorithms

The word *algorithm* has a long history (Striphas, 2015), but why was it imported into computing? In common use, the word *algorithm* can sloppily refer to a gigantic assemblage of people, institutions, computers, computer programs, strategies, and even data centers and electricity. The word is also used to encompass all the software of a large Internet platform—as in "Google's algorithm." Synonyms for *algorithm* include *recipe* or *procedure*. These uses of the term are valid, but this article focuses on the more narrow computer science definition. When they imported the concept from mathematics, computer scientists struggled to define it. In 1966, the following brief letter to the editor appeared in the *Communications of the ACM*, a leading periodical in the nascent area of academic computer science (Wangsness & Franklin, 1966, p. 243):

> We are making this communication intentionally short to leave as much room as possible for the answers.
>
> 1. Please define "Algorithm."
> 2. Please define "Formula."
> 3. Please state the difference.
>
> Signed: T. Wangsness & J. Franklin.

Two answers were provided. The first, by Hartmut Huber, defines an algorithm as "a finite sequence of rules operating on some input yielding some output after a finite number of steps," but also "any program written in [an algorithmic] language" (1966, pp. 653–654). The second answer, offered by Donald Knuth, states, "To me the word algorithm denotes an abstract method for computing some function, while a program is an embodiment of a computational method in some programming language" (1966, p. 654).

To a computer scientist, algorithms exist independently from any kind of computer, hard drive, or other physical substrate upon which they may be implemented. Even in the original framing of Wangsness and Franklin, disambiguation is sought not between what a computer does and what other mechanisms do but between algorithm and formula, two disembodied notions. Knuth's (1966) answer draws two further distinctions: between an algorithm and the function it performs and between an algorithm and the program that implements it. The latter one is of the same nature as the point above in that it separates the algorithm from its embodiment (Knuth's word), be it hardware, software, or code. Note that, to Knuth, the same algorithm can be implemented differently in two different programming languages; therefore, an algorithm is not equivalent to a program.

One can see here the tensions at work between the algorithm as an abstract set of rules or a strategy (Knuth) and the algorithm as a manifestation of such rules (Huber) in a particular algorithmic language. Between these two answers, there exists at least one level of abstraction between a computer program and its algorithmic origins, and possibly a second layer of abstraction between a problem or function and its expression in an algorithmic language suitable for bringing to a particular computational process. One remembers here that, in early computing, computing labor itself was divided between the mathematicians who devised a problem and the programmers who set up the computer to address the problem.

We will later make the case that the study of an algorithm's ethics requires examination of both its composition and its consequences. Yet the early discussion of the concept of the algorithm cited above demonstrates that, even for a seasoned computer scientist, it can be tricky to even know when you are talking about the algorithm, its implementation, is effects, or its function. Nonetheless, the word *algorithm* was useful to computer scientists because it allowed them a way to talk about a method that was independent of a particular computer or situation.

Being Specific in Our Questions About Ethics

By taking the computer scientist's view of what an algorithm is and is not, it is apparent that the question of whether an algorithm is ethical is a very focused one, because *algorithm* is a term that is narrowly defined. The question asks specifically about a certain kind of finite process and not about its goal. This is not to say that the goal is unimportant, yet asking about the ethics of the goal is simply a different question than considering the ethics of an algorithm. To evaluate the algorithm by its properties and processes requires attention to some of the specific features and characteristics of algorithms. We need to look at an algorithm's particular inputs and potential outputs and, yes, even its ethical relationship to good. To understand an algorithm as oriented to good is to trace possible effects back to causes—the actual processes at hand, which, though not reducible to effects, do reveal an algorithm's teleology and

potentially its deontological duties. Though we do not endorse a deterministic view of technological effects, we explain how an algorithm might be written in a way that tends toward or against normatively positive behavior.

Case Study: Racist Computer Vision Algorithms

Consider the problem of evaluating a surveillance algorithm as ethical in its treatment of race. In our introductory example, a photographic apparatus was found to contain embedded ideas about identifying valuable human skin colors. Moving now to a different kind of consideration of race, we present an example of a surveillance system designed to evaluate streams of surveillance camera footage in airports and train stations and decide under certain circumstances to notify a human operator that something worthy of notice may be happening. This is similar to the system described by Neyland (2016). Although our surveillance system is hypothetical, we will use our knowledge of particular, specific vision algorithms from the computing literature and cited above to proceed with this discussion.

Race is obviously fraught in the context of surveillance, public safety, and counterterrorism. Although race has been accepted as a part of the visual description of people on identity documents (Bowker & Star, 1999), even mention of the idea of race can be seen as prejudicial in the context of a screening process at an airport or train station that is not searching for a particular person. This makes race a useful test case to investigate the question of whether surveillance algorithms can be judged ethical.

Using accepted terminology often used to discuss algorithms, we will distinguish between algorithms by defining inputs and outputs. We understand the inputs in this case to be the criteria by which the algorithm selects events for possible consideration and processing, by the human operator or a computational process. We understand the outputs in this case to be the images isolated from surveillance feeds and displayed to the operator for consideration and judgment.

We understand racism in this case to be an inappropriate or unmerited consideration of race as a factor in determining possible threats to public safety, as determined by looking at either the consequences of a particular algorithm or its rules. We will imagine that the aims of the system are to avoid racist means *or* ends, and that the human operator in the system has been trained to disregard race entirely. Based on this framework, we imagine the following possible scenarios for the algorithm at hand.

Scenario 1: Race as a Predefined Skin Color

Here, we will imagine a version of the system wherein computer vision algorithms select persons for review by the operator based on skin color, as defined through hue and saturation of pixels within shapes that infer the presence of exposed skin in a human form. For example, one such algorithm normalizes lighting in images, looks for "elongated areas representing limbs and connected limbs constrained by a skeletal geometry" and then searches for skin by looking for "targets $110 < \text{hue} < 150$ where $20 < \text{saturation} < 60$ and $130 < \text{hue} < 170$ where $30 < \text{saturation} < 130$" (Fleck, Forsyth, & Bregler, 1996, pp.594–595).

Considering the consequences of this algorithm would lead us to conclude that, in some situations, the algorithm could produce racist results. It is quite plausible that this sort of algorithm could only reliably target dark-skinned people while not seeing other lighter-skinned people as people. Indeed, in the example above by Fleck and colleagues, their initial formula for estimating skin color was too heavily weighted toward light skin, and so it later had to be adjusted. Even an accurately targeted algorithm intended to retrieve all persons of light skin color could then be used for racist purposes, resulting in racist outcomes.

But we could not conclude that this would produce racist results in all scenarios. There are, for example, some situations where race is an acceptable criterion for filtering the occurrences of human forms on a live video feed. For that matter, skin color on a camera charge-coupled device is not even a consistently accurate indication of race. Skin tone effects in image processing algorithms have also been shown to be extremely sensitive to lighting—identical instantiations of this algorithm in Finland versus Singapore might lead to only one system being racist due to the quality of the natural light.

This means that we can determine that an algorithm's use of race is unethical. It produces inappropriate consideration of race when implemented in a particular system, but this requires consideration of various circumstances external to the algorithm to reach this conclusion. The algorithm, by this reasoning, cannot be racist by itself, because it is impossible to determine what the algorithm is doing without more information about its implementation and context. Not to mention that whether consideration of race is inappropriate is a judgment that could be said to depend on the application. Nonetheless, if the system as a whole is unethical and racist, at the least we can say that the algorithm helps the system achieve this status (Slack & Wise, 2005). This algorithm is designed around explicit judgments about skin tone, and that is a low-level technical detail that the ethical analyst would ignore at his or her peril.

In comparison, a different approach to applied ethics asks us to consider whether there are racist, unethical things that an algorithm might do regardless of their consequences in a particular context. This question is at the heart of race as a legal problem. In the United States, the Civil Rights Act of 1964 was seen as a major victory because it decreed that race simply could not be considered in some situations, regardless of the context, nuances, or consequences. The same color-blind logic was later used in California and Michigan ballot initiatives in 1996 to dismantle affirmative action in higher education. Of course the proponents of these measures considered the likely consequences, and they judged this rule about process would produce the consequences they desired. But they also each grappled with the argument that it might be an ethical duty to not consider race at all, because this has proven to be a potent and vexing argument in the ethics of civil rights (Bonilla-Silva, 2010).

A case could be made under this logic for forbidding the consideration of skin tone. A concerned engineer might argue that it is a bad idea to bring more consideration of race into the world. Even if this algorithm could be applied in a scenario where the effects were not racist, the scenario wouldn't matter. Even if this algorithm could be used in situations where racial discrimination was necessary and justified (such as identifying a particular person by using skin color as one of many attributes), that situation is not relevant.

Although this is not our position, various groups have seen it as a moral duty to not consider race when making judgments of any kind. This ethical position is useful to elucidate some kinds of reasoning that may be particularly salient to judgments about algorithms. Since some computer scientists define an algorithm to be about process alone, violating rules about particular processes that are forbidden is one specific way that an algorithm can be judged unethical or illegal. According to this ethical position, the algorithm is not just unethical, it can potentially be independently unethical—if any consideration of race is unethical or illegal, the algorithm can be racist on its own without considering an implementation, program, or context.

There is some logic to this even if we bring consequences back into the frame. This algorithm will inevitably select some persons based on a race they do not identify with, given the subjectivity of race as an identity category. Whenever it is applied in a situation where race is not an appropriate criterion for selection, this algorithm will always follow a racist rule. Again, the inclusion of skin tone in the Fleck et al. (1996) localizer is critical to the analysis. For a certain kind of strict ethical thinker, the appearance of skin tone in the localizer is the most important fact that can be revealed about the system.

Scenario 2: Race as a Learned Conclusion

A second kind of algorithm that might be selected for this system is a machine-learning algorithm. Machine-learning algorithms use a statistical evaluation of a set of data to develop decision rules rather than specifying all decision rules in advance. In a machine-learning process, an algorithm might begin to form conclusions based on race, and then act on those conclusions in determining which processes to apply to which events or subjects.[2]

If this machine-learning algorithm is giving weight to race—without even having a defined category for race—then the algorithm has a strong potential for racist consequences without employing any explicitly racist rules. Even though the output is not an explicit race category or variable, in effect the algorithm would be internally recognizing race because it correlates highly with the desired output. If such an algorithm were applied in a surveillance system designed to notify a human operator of suspicious activity, the operator may end up being shown only subjects of a particular race without ever knowing that such a criterion were in play. A training data set produced by a racist operator would produce a racist machine-learning algorithm. This is not necessarily an argument about consequences. Such a classifier algorithm might not start out with any racist rules, but it can acquire them later via machine learning. Once those rules exist, we can say that the algorithm considers race even if the word *race* is not used. Although this might not be foreseeable by the designer of the classifier, the person deploying the classifier could have a duty to consider the training data to ensure that race is not considered, even though it was never named. The fact that this algorithm learns from history or learns from the system's operators is critically important in judging the ethics of the system.

[2] For a list of machine-learning research relevant to this discussion, see Barocas et al. (2015).

Scenario 3: Race Without a Predefined Influence on Outputs

More speculatively, the core algorithm of a surveillance system might be revealed, it might consist of several different algorithms chained together, or each algorithm might be offered as one of several possible options to a user. How a racist algorithm's logic is exposed to the user might change our evaluation of its consequences. We might imagine a third variation of the system where a computer vision algorithm is designed to search for subjects based on race, but the algorithm only saves this information for later analysis by a second algorithm, an internal or external process. Such a system might even present this information for possible use by the operator, pending a particular event or input. In such scenarios, where race is available to an algorithm as an input but not clearly connected to outputs, we can see potential for racist consequences, but perhaps less potential. A classifier algorithm could, in other words, return images to the surveillance system operator already sorted, or it could return to the operator a choice of possible ways to sort the data. The latter would lessen the possibility of racist results.

Three Justifications for a Good Algorithm: Virtue, Consequence, Norm

We mean our reasoning about algorithms to be a practical endeavor, because "the algorithm" as an idea has captured the scholarly imagination at this moment because of its actual implementation in real technological systems whose consequences are felt daily. This discussion intends to address critics, scholars, and analysts, but also algorithm designers and system builders with applied problems. However, practical ethics is an impractically messy place where hypotheticals and abstractions abound. We have said that focusing on the algorithm is an ethically productive strategy. To conclude our analysis, we need to clarify what we mean by asking whether an algorithm is ethical, because various forms of ethical reasoning should be considered.

This section will begin with some elementary examples. The point here is not to approach ethics abstractly as a philosopher might, but to investigate the kinds of reasoning about ethics that people actually do. For instance, although they may not be aware of it, each member of the Association for Computing Machinery (ACM), by its own account the world's largest educational and scientific computing society, is required to abide by a code of ethical conduct (Association for Computing Machinery, 1992). The ACM Code, like most efforts in applied ethics, attempts to grapple with moral problems by borrowing tactics from every major branch of the philosophy of ethics.

ACM members are required to abide by three very different kinds of ethical precept. They are asked to "be honest" (1.3), a form of virtue ethics traceable to Aristotle and grounded in character. Although no computing equivalent of Hippocrates has yet emerged, ACM members are required to "avoid harm" (1.2), a form of teleological ethics asking members to reason about ends and not means. Indeed, they are explicitly asked to "assess . . . social consequences" (1.2), easily recognizable as consequentialism. Finally, ACM members are told to respect copyright law (1.5), a deontological ethic grounded in the idea of following a predefined set of moral rules because of those rules.

These are three different kinds of justification, but all three handle the same questions of morality. We could abstain from lying because it is a part of our character to be honest (virtue ethics),

because lying produces some harmful consequence (consequentialist) or because there is a norm that forbids it, regardless of the situation (deontological). In real-life problems, such as the design of surveillance systems, people typically proceed using a hodgepodge of all three approaches. Although these three kinds of reasoning are logically incommensurate when taken strictly, in practice they are not interpreted strictly. Our conclusion in this article is that it is possible and useful to perform an ethical assessment of an algorithm. With respect to the ACM Code, this moves an analyst's focus away from the computer scientist and into the computer. If we consider an algorithm to be an independent agent (Callon, 1986), we find that these three ethical approaches provide varying degrees of analytic traction.

Algorithms With Vices and Virtues: A Difficult Proposition

Why should we expect a feature-based face localization algorithm to be honest or exhibit any other kind of virtue? And yet the computer science jargon is full of references like this. In computing, an allocation algorithm is said to be fair if gives access to different flows of data in a round-robin fashion. Another is greed: A greedy algorithm follows logical paths of higher perceived value first. Likewise, a vindictive networking algorithm punishes nodes that have used more than their share of bandwidth in the past. We have not encountered an algorithm that is characterized as modest, courageous, or pure, but it should not be ruled out.

These characterizations are used as a way to describe some of the basic techniques underlying an algorithm rather than as bona fide ethical judgments; therefore, computer science and virtue ethics quickly part ways when these ideas are considered more than superficially. Virtue ethics may not be appropriate for the analysis of algorithms, because a virtue ethic represents a disposition rather than a rule. An honest algorithm in virtue ethics would be one that behaves honestly in a wide variety of situations, some of them difficult to foresee. A virtuous person is not a person who rigidly applies a rule regardless of the context; indeed, to be rigid is a vice. Even if we accept that an algorithm can have a disposition or a character, intentionality or mind-set is the key component of virtue ethics, and while we may grant that a nonhuman algorithm has agency and power, granting it intentionality (or a mind) seems a bridge too far (Latour, 2005).

This means that when we describe an algorithm as fair we probably mean to reference either its consequences or some absolute norm of fairness, and not that the algorithm itself has innate virtue. Virtue ethics as an avenue of reasoning about algorithms does not seem particularly plausible.

Consequentialism and Context: The Easiest Path

We are then left with consequentialist and deontological ethics. Consequentialist ethics asks us to turn to the results in a particular context to judge morality—it asks us to consider consequences. Considering the ethics of an algorithm by considering its consequences demands that we trace its instantiation in actual situations, or at least imagine them. We must therefore understand the web of associations and interconnections that algorithms have with other technologies and people. Neyland's (2016) call to always see algorithms as situated in context, referenced earlier in this article, is, then, a call for a consequentialist ethics of algorithms. Because the same algorithm might have very different

consequences in different situations, this might seem to present a problem for pursuing any ethical reasoning that focuses on the algorithm itself.

Luckily, this challenge is easily surmounted. While indeed the same algorithm might have different consequences in different situations, the same algorithm might also have the same consequences. That is, there may be broad trends in the consequences of certain algorithms, and it may be crucial for society that we detect them. In the scholarship that is normatively critical of particular algorithms (Gillespie, 2012; Introna & Nissenbaum, 2000), there is no sense that these authors ever expect the algorithm to operate independently. A focus on the algorithm does not preclude attention to the context, nor does it substitute for nuance. In fact, this critical movement is arguing that within the nuanced technical details of an algorithm, we find important political consequences that have not received public attention commensurate with their societal importance (Introna & Nissenbaum, 2000). No doubt the recognition that every technical system operates within an interconnected and complex web of context is a penetrating diagnostic strategy, but we emphasize that, in addition, it is a useful strategy to learn to read and investigate algorithms as a particularly prevalent and potentially significant component of our evolving infrastructures.

A short side example will help to cement the consequentialist case and also explain that focusing on the technical details of the algorithm is not inconsistent with considering context, systems, and consequences more broadly. Crain (2013, 2016) considers the late 1990s evolution of two broad classes of algorithms that are intended to determine which advertisements to display to particular users on the Web. He explains that *profile-based* algorithms employed by DoubleClick attempted to uniquely identify particular users, linking their online activity to their off-line life. Once this link was established, DoubleClick would then mine a user's actual demographics and past purchases for intent to purchase certain products and display related ads. In comparison, *interest-based* algorithms employed by Google attempted to mine the keyword search history associated with a particular Web browser to infer purchase intentions.

DoubleClick's *profile-based* algorithms relied fundamentally on amassing immense databases of uniquely identifying information, giving DoubleClick's approach far more potential for some forms of negative consequences than Google's *interest-based* algorithms. These possible consequences include an increased risk of identity theft, the invasion of privacy, and civil rights violations. But understanding the algorithm in this case does not preclude a broader perspective: In fact, it requires one. Crain points out that the decision to use one or the other class of algorithm was a business strategy that depended almost entirely on a company's access to inputs. These inputs were either large databases of personal information or large databases of keyword search history (Crain, 2013).

The distinction between the consequences of the two kinds of algorithm was significant until Google acquired DoubleClick in 2008. Advertising companies then merged both approaches, in part by using corporate acquisitions to secure the inputs for their competitor's algorithm. Although the industry no longer distinguishes between profile and interest, it remains valuable to understand the technical process by which advertisements are selected, because the current hybrid approach retains the worst dangers of its predecessors. Understanding the business context is certainly important in Crain's analysis, but so is

understanding the details of two kinds of specific algorithms. Indeed, the business strategy does not make sense without them. Crain does not need to resort to considering algorithms as immutable, isolated, or without context.

Of the three kinds of ethical reasoning considered here, consequentialist ethics seems the most productive way to think about algorithms, and it is an intuitive way to think about an ethical problem, akin to asking, "What will happen if we do it this way?"

Thinking Deontologically: Locating Algorithmic Norms

The third form of ethical reasoning—a deontological ethics of algorithms—feels like a counterintuitive proposition. In deontological terms, honesty is neither the character trait "honest" nor a decision made after considering the consequences of lying. Instead, it requires the development of a series of moral and potentially even absolute rules ("do not lie" or "do not duplicate copyrighted software without permission") that can be followed. The word *deontology* itself is derived from the Greek word for "duty." Most people hold ideas that are likely justified deontologically because they don't necessarily depend on consequences, and these are widely accepted as reasonable. A deontological ethics of algorithms feels unpromising largely because the deontological precepts that are easiest to grasp without resorting to consequentialism seem unlikely to appear in the realm of the algorithm. Therefore, analysts have commonly turned to consequentialism.

We contend that it is an interesting and open intellectual question whether a set of norms exists or could exist for algorithms. Applied ethics in real-world settings typically incorporates both rule-based and consequences-based reasoning. Indeed, as we explain below, it can be very useful to explicitly rule out some kinds of algorithmic process a priori. We may not have such norms now, but that may be because automating contemporary situations with computer algorithms is a new practice. Just as it is often held that torture is wrong regardless of the situation, or certain human rights are absolute, having an absolute rule for algorithmic behavior does not foreclose our also reasoning about consequences in addition to specifying some norms or rules in advance.

To consider algorithms deontologically would be to say that they must comply with a set of norms or moral duties. This is intellectually interesting in part because it has the most potential to provide guidance for algorithm designers of any approach we have considered. In this framing, the command "Comply with published protocols!" could be ethical guidance as well as practical engineering advice. An algorithm for sending spam could then be unethical regardless of whether it successfully sends spam, and irrespective of the consequences of spam. It could be judged unethical deontologically because it performs e-mail spoofing in violation of the SMTP protocol that governs e-mail. While most rules in the SMTP protocol appear there to make the overall system of e-mail function correctly, other rules and explicit reference in the protocol to "spammers" have a moral valence and seem justified by the fact that some messages are desirable and others are undesirable. This is a function of community norm rather than efficiency (Brunton, 2013).

In general, although reasoning deontologically about algorithms seems intuitively difficult, this is an area of interesting work where little analysis has been attempted so far. It is our conclusion, as we hope the examples in this article have demonstrated, that deontological ethics of algorithms should not be discounted and that some a priori ethical algorithmic norms might be possible. In the case of a hypothetical surveillance system described above, the use of a judgment about race in a surveillance algorithm is already part of a broader public moral conversation about how we should think about race. To absolutely forbid race in the process of making college admissions decisions and in the process of surveillance are both deontological positions.

Conclusion: Recentering the Algorithm, Narrowly Defined

We have argued that is essential to be essentialist about algorithms. Algorithms represent a fruitful path of normative investigation, with important dynamics and consequences that could deserve their own dedicated analytical vocabulary. In this article, we have often implied that algorithms have fixed characteristics. We believe this can be a powerful analytic strategy, and indeed it is a crucial one if we are to engage as normative analysts of contemporary technological infrastructure so that we have some handhold on what it is we are investigating and arguing about. In this article, we have construed the term *algorithm* narrowly and specifically, to relate it to ethical reasoning, and we have used some specific computer vision algorithms from the computing literature in the context of race and surveillance in an attempt to demonstrate that it is possible and hopefully practical to ask ethical questions by focusing on algorithms. Even though they are often presented as too technical, too black-boxed, or too "low-level" processes in computing to sustain scrutiny, many algorithms are in fact widely available in computer science textbooks and are understandable by computer programmers. While we agree that some complicated algorithmic systems may be unknowable in practical terms (Seaver, 2013), other algorithms are in fact published, legible, and comprehensible. Assuming that all algorithms cannot be scrutinized because they are unknowable is silly.

To conclude, we return to the general role of algorithms rather than particular instantiations or kinds. Our infrastructures are increasingly networked and dominated by computation. The remarkable new prevalence of algorithms and the widespread claims for their significance are reason enough to investigate their consequences. During this moment, there are three reasons we need to study algorithmic ethics.

The Increased Delegation of Authority

First, as they are implemented today, algorithms have a remarkable ability to delegate authority. Technology itself could be described as a way to delegate authority or control (Pinch & Bijker, 1984), but even beyond this generality for all technology, the algorithm has fostered this delegation to great effect. A remarkable amount of apparatus has emerged in the last 50 years to standardize and disseminate algorithms and programs (from computer science education to GitHub). This apparatus makes algorithm implementations into interchangeable parts and makes them seem like a black box. This means that algorithmic design, intent, action, and consequence can be distant in space and time. When authors describe algorithmic infrastructure, they intend to highlight systems that operate with algorithmic logic,

but the infrastructure of algorithms is just as important because it leads to important changes in how our sociotechnical systems function.

As we have argued that the details of algorithmic process are important, we have intended to highlight that choosing one algorithm over another in order to address the same problem may be a decision of significance. So might the manner in which an algorithm is implemented. Of course, circumstances also constrain the choice of algorithm, or even its design—as Google pushed for better algorithms to infer purchase intent from search keywords, because as a search engine company, it had ready access to a vast trove of search engine keyword data. Due to this delegation, an algorithm developed in a distant country may be the site of inquiry that a thorough ethical analysis of a technical system demands.

The Rise of Ongoing, Networked Control

In our opening example, we contrasted HP's "racist" face localization algorithm with the early history of photography and cinema. We pointed out that Dyer mounted an effective critique of photographic technology without the concept of algorithms. Although the term *algorithm* did not yet exist, Dyer's cameras presumably had algorithms in the sense that they had processes, if not in a formal, narrow sense. These processes could be, for example, chemical. Yet it is today's implementation of algorithms in networked computers that asks us to reconsider our methods in the analysis of technology.

The distribution and reconfiguration of algorithmic processes implemented in computers can be much faster than most previous approaches to distributing control, logic, or organizing process in technology design. This is one of the reasons computation is exciting: The fact that our newer technologies are networked and reconfigurable is widely seen as a benefit. As Steiner (2012) points out when discussing catastrophic examples of automated trading algorithms used by Wall Street, an algorithm's consequences can also be extremely rapid. The HP face localization problem was changed via a software update that modified the operation of thousands of cameras that had already been deployed, and this software update was deployed within a few weeks.

Although an algorithm does not necessarily have to be instantiated in a way that is easy to update, the fact that this capability is possible changes the functioning of our technological apparatus profoundly. The metaphor of the technology's diffusion changes from that of an inventor hopefully sending products out into the world to that of puppeteers whose movements remain always connected to their performances.

Stabilization and the Algorithm's Differential Obscurity

In the study of science and technology, stabilization has been an important concept used to explain the evolution of technologies and ideas over time (for a review, see Humphreys, 2005). It refers to the point at which a single technological artifact becomes the default representation of that kind of artifact in the minds of either technology developers or the public. The concept of stabilization developed, in part, from scholars of science who tried to understand how scientists determined which areas of work contained

important unsolved problems and which were closed. Although the term was meant to refer to physical artifacts in technology studies, we can see from the discussion above that it is an important analytical question to examine the degree to which an algorithm has been stabilized. The ease or difficulty of that stabilization is another relevant area of inquiry.

To explain, some technologies appear to employ the potential reprogrammability of the algorithm (such as the Google search algorithm or the Twitter "trending topics" algorithm) to make a key feature of their system modular and subject to continual revision. While modularity has been a critical concept in the history of technology since the industrial revolution, if software were explained using the metaphor of a car, the people at Google and Twitter do not just want to be able to use different tires in winter or obtain transmissions from a different supplier; they want to change the engine every day. This has focused attention on the plasticity of algorithms, as mentioned in the previous point, and suggests a kind of permanent destabilization for some algorithms.

Scholars of these algorithms have called them opaque, but this refers to corporate secrecy and the complexity and unfamiliarity of math and computer code. Yet this plasticity is only one possible use of algorithmic logic, and this is only one kind of opacity. Recalling our earlier point about the distribution of authority, imagine, for instance, that the Viola-Jones face detection algorithm becomes the accepted, normal algorithm for detecting the presence of faces. We know that algorithms often contain other algorithms, and face detection is only one step among many for a surveillance application. This presents the possibility that an ethically problematic algorithm, once stabilized or widely accepted as the default solution for a particular problem, becomes encapsulated in an encapsulation. It can be buried so deeply in the logic of a system that it might be very difficult to detect, even for engineers working on the system, who might not know that they use it.

While the role served by that particular localization algorithm might be made visible and it might be made modular, it can also be designed into expensive-to-change hardware, buried, and forgotten. This again argues for the relevance of algorithmic skills that allow a facility with the relevant ideas in math and computer science as well as the education of computing professionals in social science and ethics. At base, this may not be a different tactic than the cinema historian (Dyer, 1997) learning about the chemistry of photographic processes. Yet saying that today's ethical critic needs a facility with these tools of computing imagines a different kind of scholar who is able to bridge the social and technological.

References

Ananny, M. (2016). Toward an ethics of algorithms: Convening, observation, probability, and timeliness. *Science, Technology and Human Values*, *41*(1), 93–117. Retrieved from http://doi.org/10.1177/0162243915606523

Angwin, J. (n.d.). The What They Know series. Retrieved from http://juliaangwin.com/the-what-they-know-series/

Association for Computing Machinery. (1992). ACM code of ethics and professional conduct. Retrieved from http://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct

Barocas, S., Friedler, S., Hardt, M., Kroll, J., Venkatasubramanian, S., & Wallach, H. (2015). FAT ML resources: Fairness, accountability, and transparency in machine learning. Retrieved from http://www.fatml.org/resources.html

Bermejo, F. (2007). *The Internet audience: Constitution and measurement*. New York, NY: Peter Lang.

Bonilla-Silva, E. (2010). *Racism without racists: Color-blind racism and the persistence of racial inequality in the United States*. Lanham, MD: Rowman & Littlefield.

Bowker, G. C., & Star, S. L. (1999). *Sorting things out: Classification and its consequences*. Cambridge, MA: MIT Press.

Brunton, F. (2013). *Spam: A shadow history of the Internet*. Cambridge, MA: MIT Press.

Callon, M. (1986). *Mapping the dynamics of science and technology: Sociology of science in the real world.* Basingstoke, UK: Macmillan.

Chetverikov, D., & Lerch, A. (1992). Multiresolution face detection. In *Proceedings of the 6th Workshop on Theoretical Foundations of Computer Vision* (pp. 14–21). Berlin, Germany: Akademie-Verlag GmbH.

Crain, M. (2013). *The revolution will be commercialized: Finance, public policy, and the construction of Internet advertising*. Champaign: University of Illinois Press.

Crain, M. (2016). The limits of transparency: Data brokers and commodification. *New Media & Society*, 1–17. Advanced online publication. Retrieved from http://doi.org/10.1177/1461444816657096

Diakopoulos, N. (2015). Algorithmic accountability. *Digital Journalism*, *3*(3), 398–415. Retrieved from http://doi.org/10.1080/21670811.2014.976411

Dyer, R. (1997). *White*. New York, NY: Routledge.

Fleck, M. M., Forsyth, D. A., & Bregler, C. (1996). Finding naked people. In *Proceedings of the 4th European Conference on Computer Vision* (Vol. 2, pp. 593–602). London, UK: Springer-Verlag.

Floridi, L., & Sanders, J. W. (2004). On the morality of artificial agents. *Minds and Machines*, *14*(3), 349–379. Retrieved from http://doi.org/10.1023/B:MIND.0000035461.63578.9d

Galloway, A. R. (2006). *Gaming: Essays on algorithmic culture*. Minneapolis: University of Minnesota Press.

Gangadharan, S. P. (Ed.). (2014). *Data and discrimination: Collected essays*. Washington, DC: New America Foundation. Retrieved from https://na-production.s3.amazonaws.com/documents/data-and-discrimination.pdf

Gillespie, T. (2012, March). Can an algorithm be wrong? *Limn*. Retrieved from http://limn.it/can-an-algorithm-be-wrong/

Goffey, A. (2008). Algorithm. In M. Fuller (Ed.), *Software studies: A lexicon* (pp. 15–20). Cambridge, MA: MIT Press.

Graf, H. P., Chen, T., Petajan, E., & Cosatto, E. (1995). Locating faces and facial parts. In *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition* (pp. 41–46). New York, NY: IEEE.

Grimmelmann, J. (2008). The Google dilemma. *New York Law School Law Review*, *53*(4), 939–950.

Hamilton, K., Karahalios, K., Sandvig, C., & Eslami, M. (2014). A path to understanding the effects of algorithm awareness. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (pp. 631–642). Toronto, Canada: Association for Computing Machinery. Retrieved from http://doi.org/10.1145/2559206.2578883

Hannak, A., Sapiezynski, P., Molavi Kakhki, A., Krishnamurthy, B., Lazer, D., Mislove, A., & Wilson, C. (2013). Measuring personalization of Web search. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 527–538). Rio de Janeiro, Brazil: Association for Computing Machinery. Retrieved from http://doi.org/10.1145/2488388.2488435

Huber, H. (1966). Algorithm and formula: Letter to the editor. *Communications of the ACM, 9*(9), 653–654.

Humphreys, L. (2005). Reframing social groups, closure, and stabilization in the social construction of technology. *Social Epistemology*, *19*(2–3), 231–253. Retrieved from http://doi.org/10.1080/02691720500145449

Introna, L. D., & Nissenbaum, H. (2000). Shaping the Web: Why the politics of search engines matters. *Information Society*, *16*(3), 169–185. Retrieved from http://doi.org/10.1080/01972240050133634

Knuth, D. (1966). Algorithm and program, information and data: Letter to the editor. *Communications of the ACM, 9*(9), 654.

Latour, B. (2005). *Reassembling the social: An introduction to actor-network-theory*. New York, NY: Oxford University Press.

Neyland, D. (2016). Bearing account-able witness to the ethical algorithmic system. *Science, Technology and Human Values*, *41*(1), 50–76. Retrieved from http://doi.org/10.1177/0162243915598056

Neyland, D., & Möllers, N. (2016). Algorithmic IF . . . THEN rules and the conditions and consequences of power. *Information, Communication & Society, 19*(0), 1–18. Retrieved from http://doi.org/10.1080/1369118X.2016.1156141

Pasquale, F. (2015). *The black box society: The secret algorithms that control money and information*. Cambridge, MA: Harvard University Press.

Pinch, T. J., & Bijker, W. E. (1984). The social construction of facts and artifacts: Or how the sociology of science and the sociology of technology might benefit each other. *Social Studies of Science*, *14*(3), 399–441. Retrieved from http://doi.org/10.1177/030631284014003004

Sandvig, C. (2014). Seeing the sort: The aesthetic and industrial defense of the algorithm. *Media-N*, *10*(3), 35–51.

Sandvig, C., Hamilton, K., Karahalios, K., & Langbort, C. (2014). Auditing algorithms: Research methods for detecting discrimination on Internet platforms. *Pre-conference on Data and Discrimination at the 64th annual meeting of the International Communication Association* (pp. 1–23). Seattle, WA: International Communication Association.

Seaver, N. (2013). Knowing algorithms. In *Media in Transition 8 International Conference* (pp. 1–12). Cambridge, MA: MIT.

Seaver, N. (2014, January 28). On reverse engineering: Looking for the cultural work of engineers. *Medium.* Retrieved from https://medium.com/anthropology-and-algorithms/on-reverse-engineering-d9f5bae87812#.xje7c8cd4

Simon, M. (2009, December 22). HP looking into claim webcams can't see Black people. *CNN*. Retrieved from http://www.cnn.com/2009/TECH/12/22/hp.webcams/index.html

Slack, J. D., & Wise, J. M. (2005). *Culture and technology: A primer*. New York, NY: Peter Lang.

Steiner, C. (2012). *Automate this: How algorithms came to rule our world*. New York, NY: Penguin.

Striphas, T. (2015). Algorithmic culture. *European Journal of Cultural Studies*, *18*(4–5), 395–412. Retrieved from http://doi.org/10.1177/1367549415577392

Totaro, P., & Ninno, D. (2014). The concept of algorithm as an interpretative key of modern rationality. *Theory, Culture and Society*, *31*(4), 29–49. Retrieved from http://doi.org/10.1177/0263276413510051

Wangsness, T., & Franklin, J. (1966). Algorithm and formula. *Communications of the ACM*, *9*(4), 243. Retrieved from http://doi.org/10.1145/365278.365286

Yang, M. H., Kriegman, D. J., & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(1), 34–58. Retrieved from http://doi.org/10.1109/34.982883

Zamen, W. (2009, December 10). HP computers are racist. *YouTube.* Retrieved from https://www.youtube.com/watch?v=t4DT3tQqgRM

Ziewitz, M. (2015). Governing algorithms myth, mess, and methods. *Science, Technology and Human Values, 41*(1), 3–16. Retrieved from http://doi.org/10.1177/0162243915608948