EPISODE 1403

[INTRODUCTION]

[00:00:00] KP: Charlie Gerard is an incredibly productive developer. In addition to being the author of *Practical Machine Learning in JavaScript*, her website charliegerard.dev has a long list of really interesting side projects exploring the intersection of human computer interaction, computer vision, interactivity and art. In this episode, we touch on some of these projects and broadly explore how practical it is to bring interesting HCI concepts into one's work.

[INTERVIEW]

[00:00:28] KP: Charlie, welcome to Software Engineering Daily.

[00:00:31] CG: Thanks for having me.

[00:00:32] KP: Where'd you get your start in software?

[00:00:35] CG: I don't actually have a background in computer science. I studied marketing at the time, but then, when I started my career in marketing, I realized that I kind of wanted something a bit more challenging and I was working with developers at the time. I did a coding bootcamp about seven years ago now, when they were not very trendy. And yeah, I learned to code over there. I did like 12 weeks intensive boot camp and then I started my career from there. A little bit of a career changer, but I'm happy that it's now a possibility for people to get into tech.

[00:01:05] KP: What was your first language?

[00:01:07] CG: At the time, the backend was in Ruby and Rails. In the frontend, it was HTML and CSS. What was trendy at the time was Angular 1. Very different from the current versions of Angular. But React started to be a bit popular, but I think people at the time still found it a bit weird. Now, we can see that things have changed a lot.

[00:01:28] KP: Are there any frameworks you've gravitated towards?

[00:01:32] CG: Since I started my career, I've mostly been using React. I did try a little bit of other frameworks. At some point, I was working on a project that was using Ember.js, but not many people use it still, or at least I haven't really encountered it on projects. It's definitely not my favorite. But I've mainly focused on React since then in the frontend.

[00:01:52] KP: What was your first exposure to TensorFlow?

[00:01:54] CG: At the time, I was actually using it in Python. I think machine learning was on my list of things that I wanted to learn for a while. Actually, quite early on, I think in my bootcamp, people started talking more and more about machine learning. I was, I really want to see what that's like, but I felt like it was something that was completely out of my reach. But then over time, on my spare time, I was like, "Oh, I'll do a little course online." I did a bit of that in Python. Then I think, TensorFlow.js came around and I was like, "Oh my God! I already know JavaScript, so now I don't have to learn Python on top of machine learning. I can use my JavaScript knowledge and only focus on learning the machine learning part." I think that was maybe like four or five years ago, trying to like experiment a little bit with this. Then I really liked it, and I came up with more ideas. And yeah, led me to working with it quite a lot over the past few years. Always on side projects, never really on the job, but still –

[00:02:53] KP: Well, if a person wanted to, they could probably spend a decade studying machine learning. There's so much academic components to it, all very valuable. But having taken a different path, do you have a view on what's the minimum viable machine learning path?

[00:03:07] CG: I think to me, well, there's a lot of resources around. But when people are getting started, I think what they really need to know is that you don't have to – it's not really about building your own algorithms. Because a lot of this is already done. You can think of it a bit like computer science, where you learn to implement algorithms, but you don't do the work to actually come up with a new one that has never been done before. It's similar in that way, where it's about understanding which algorithm are good for what purposes, and then working on obligations and building projects, so that you understand how to put that into practice, and what it can do, what it can't do. I mean, that's how I do it. I don't know if this is how everybody does it.

But I feel like at the time, when I got started, I didn’t know if it was all about, math and building your own algorithm, but it wasn’t really about that. A lot of the times, you have open source models that are available and it’s about using it in your app and preparing the inputs for the model and getting predictions out of that. But it’s not really about the actual algorithm that you have to create yourself. It’s not really about that, at least when you start.

[00:04:18] KP: I think most people who’ve worked with TensorFlow found their way to it through Python, that seems to be the most popular path. With it comes some staples of the Python ecosystem, like Pandas and NumPy. Are there any equivalent on the JavaScript side or can you compare and contrast the developer experience?

[00:04:35] CG: There’s definitely more tooling in Python, because in general, production applications are made in Python and not really in JavaScript or from my experience. In JavaScript, in general, what I do is more – using a pre-trained model or builds my own model. But in terms of visualization, I think there’s some playgrounds that Google open sourced as well, but I wouldn’t know the rest of the – that’s not really like the part that I focused on my side project. I wouldn’t know right now, but maybe that’s the space where more tooling can be created if people are interested in doing that.

[00:05:08] KP: Definitely. What does it take to get up and running, doing some image classification?

[00:05:14] CG: What is good is that if you use a pre-trained model, you can actually do that in just a few lines of code. In general, that’s what I would recommend if you’re getting started. Because I think it would be unnecessary to try to start everything from scratch, when you’re trying to get to know what machine learning is about and how to create things in the frontend. I would always rely on tooling that’s already there. You have for example a mobile net model or the Coco SSD model that are machine learning models for image detection, and you can import that in your frontend. You give it either an image or data from the live feed of a webcam and it does the prediction for you. In a few lines of JavaScript, you can get started actually pretty easily. Then you can build up complexity if you want to afterwards.

[00:05:59] KP: I can get TensorFlow.js from NPM install. Where can I get the pre-trained model from?

[00:06:06] CG: There's a repository on GitHub as well, I believe. On the TensorFlow.js repository on GitHub, I think you have different folders, and you have a folder for pre-trained models where you have some for image detection, or a text classification, you have some for face detection and hand detection with the webcam as well. All of this, usually, I find them on the GitHub repository of TensorFlow.js.

[00:06:30] KP: What does this look like if I want to deploy it? How do I get some model into production?

[00:06:35] CG: Personally, I have not done that. I've only done my side projects. There's a few different ways. But personally, like as I've only done side projects, I wouldn't know for sure what like the best way would be. I think the thing to remember more specifically is that, if you do deploy a model for production, a lot of the times you have to think about the fact that it will have to be retrained over time. Because as you feed new data into the model, you have to build a kind of like an architecture or a system that would allow you to retrain your model and then re-upload it. It's something that you have to think about. But in terms of ways to actually deploy your model, that's not my area of expertise.

[00:07:16] KP: What about ways of knowing when your model is ready, either to show someone or maybe to think about putting in production? How do you look at the maturity of what you've built?

[00:07:26] CG: There are certain frameworks. I forgot the name of it, but I think Google has certain guidelines around knowing like – let's say the fairness detection of your of your model, for example. I do a lot of testing manually of like – I give it an input, and does the output makes sense. But it's also about the data that I feed it, is it diverse enough, does it contain a lot of different types of input. It would depend on what you're working on. But for example, if you wanted to do face detection, you would want to make sure that the people that are part of your input data are diverse, so they actually cover the same type of people as the world has. Just in terms of diversity in general, and you would have to make sure that the predictions that come out of your model are accurate.

In general, for any model that you use, you would calculate percentage of accuracy and you want to get it as close to one as possible. In general, it comes back between like zero and one, or zero to 100 if you're in percentage. You would want to get it as close to 100% as possible, but then it depends on what you're building as well. But it's through trial and error, you change some parameters in the way that you're training your model. It's called like hyper parameter tuning. That is more like an art than a science. You can't know in advance the exact parameters that you have to use to get a super accurate result. It will depend on the amount of data that you have, on the type of data, on the application. It's not really like in JavaScript, when you write something and you know what's going to happen. You have to do a lot of tuning before you get there.

[00:09:02] KP: Well, can you tell our listeners a little bit about *Practical Machine Learning in JavaScript*, the book? What are some of the general overviews of the contents?

[00:09:10] CG: Sure. What I tried to do with this book is I wanted to make it as clear as possible for people getting into machine learning. I know that personally, for me, before, I got into it, I was quite worried that I would not understand at all what it is because I felt like it was really different from frontend development. The goal with this book was to make it easier for people who have a JavaScript background to get into it, and to show that it doesn't have to be something that's out of your reach. You can start with a JavaScript that you already know. It's about importing models, and building applications. I gradually tried to go into complexity with the book, and I also want to cover the different types of applications that you can build with it. A lot of the times, we do talk about object detection. I think I start with that, but there is also a text classification and how you could build bots if you wanted to, or you can – then I go a little bit more complex into using data from hardware. For example, with the Gyroscope and Accelerometer in your phone and building your own machine learning model. So yes, I tried to make it a bit less scary. So that by the end of the book, people would understand more what's possible on the web with machine learning. And hopefully, from there, people can get into more complicated topics. I would hope to provide kind of like the entry point for people who are interested.

[00:10:33] KP: There's a lot of things a frontend web developer can be learning about. They got to pick how to spend their time. They might look at something like a relational database and say, "That's part of the backend. I'll connect to it through an API. I don't need to learn SQL and relational databases." They might have a similar opinion about machine learning that these are these big models trained on the back end and batch processes. Not so important for a web developer. What can a web developer do with machine learning?

[00:11:01] CG: Well, personally, because I liked it and I'm looking into it. I think there's a lot. I mean, in production applications, I don't think I've seen so many things. But there are definitely, I think the classic use cases are around image detection. I think a lot more could be done. I think there's maybe still this aspect of thinking that it's only – it should only be done in Python. Whereas, well, personally, I don't think it's true, but I haven't really seen that much adoption in the frontend yet or at least not as much as I would like to. But a lot of the times, even when I might work, I'm trying to think about, "Oh! Could I add some machine learning onto Netlify?" But I think there are some applications around predictive prefetching, is one that at some point was a bit popular, maybe a couple of years ago around using the data that you get from your users on Google Analytics and looking at – like trying to predict where people are going to click next, so that you can prefetch these assets for performance.

That's definitely something that I wanted to look into. I didn't really do it, because I didn't have access to the analytics. But that's one application that I think could be really interesting in terms of performance. But also, I was interesting a little bit in creating interactive experiences around text detection, and maybe building more like smarter bots. At some point, bots were very trendy as well. But sometimes, they don't quite – the way you interact with them, you don't really get the answer that you want. With the data that we get from users, I really feel like we could do more with texts. There were some applications around toxicity classification, especially with how people use social media more and more. There's so much content that's actually available that we could be training models around detecting potential toxic content. But yeah, there's a few different things that can be done.

[00:12:51] KP: I really enjoyed looking through some of the latest projects you have posted on your website. Could we start with just an overview? What are some of the areas you're interested in developing?

[00:12:59] CG: Sure. When I start with project, I never really think about, is it going to be useful or not. It's more about exploring different ways of using machine learning. Because I think when you get started with it, a lot of the times, the hello world of it is around detecting handwritten digits on images. That's usually like the start point. But then, I know that for me to be excited with projects, I have to push it a bit further and seeing like, what else can I do? Because as it's something that I do on my own time, I want to make sure that I keep getting excited about it. I need to like make it more exciting for me at least. I've been looking a lot lately around movement detection with the pose net or movement TensorFlow.js model. Because I'm interested in exploring different ways of interacting with technologies and with the web, rather than sitting in front of my keyboard and just typing away. I feel like we could make experiences a lot more interesting with other ways of interacting. With using things like skeleton detection with pose net, you can use all of these key points that make your body with like the shoulders, or the wrists or whatever to create other interfaces, either games or ways to manipulate an interface with something else than the keyboard. That's usually what I try to get into, like it's called human computer interaction. Usually, I try to find an idea around that space because that's what I like and then apply it to different technologies.

[00:14:28] KP: Do you consider yourself more of an HCI researcher or would you say digital artists as inappropriate label?

[00:14:35] CG: Oh! I like both.

[00:14:38] KP: Yeah? We'll do both then. Well, let's take it on maybe the *Squid Game's* body movement detection project you've done. For listeners, can you describe what it is?

[00:14:48] CG: Sure. If you watched *Squid Game* on Netflix or actually, if you haven't, I'll explain quickly. This series at some point has, in one episode has a game where you have like a bunch of people who run and there's this big doll, I'm going to say like that, that you can run for a few seconds. And then if the doll is looking at you, you have to freeze. And if you move, when you're not supposed to move, then you die. With TensorFlow.js, as I know that I can recognize the body in the webcam feed. I decided to repurpose a small project that was already done by somebody else, but I did pose detection to it.

Using the webcam feed, I kind of play this game where when I can move, I get closer to my computer. But when I'm not supposed to move, then I do some calculation in JavaScript to check that the – let's say, how am I going to explain that? The delta between the position of different points in my body is not moving too much. If I move, then I fail in the game. If it detects that I'm not moving, if I'm playing the game properly, then it lets me do another round until I get to my computer. I only tested it with myself, but the PostNet's model, I think can detect multiple people as well. There's probably a limit. I don't know how many can detect at a time. But it could also be something that you could play with multiple people, and it would do multiple people detection. But I haven't actually tested it. I think I only wrote the code for myself. So, yeah.

[00:16:15] KP: Could you talk a little bit more about the implementation? The part that's kind of fun for me is the function that gets called and is passed in all the pose data. But of course, that's part of this real time streaming system. How do you get the whole thing going?

[00:16:29] CG: Sure. To get the data from the webcam, I'm using the web API, the getUserMedia web API. That's something that's probably already used in web platforms when you're making calls or something like that. That's not related directly to machine learning, but I'm getting the data from the webcam like this. That is then fed into the machine learning model that I have to load first, and then it gets the data through the video element. Out of that, when a pose is detected, I get a big object with all the points from my body that it can detect. I forgot how many it is. Maybe 16 or something like that. Each of these points, for example, the right shoulder, I would have an X, Y and Z coordinate and then you can use that in the browser as you would do on canvas. That's why you can paint if you want. It's all about positioning in the browser.

Then you can do some calculation, because using the coordinates, you can detect movements, or you can link two points together. So you can say, "Okay. If the coordinates of my right hand is between 0% and 50% of the screen, for example, then I'm raising my hand. Or if the coordinate of my hand is above the one of my shoulder, then it means my hand is over my shoulder and things like that." You can kind of create your own interactions that you're thinking of. At some point, at the beginning of the pandemic, I created an experiment that was detecting when I was running in front of my screen to trigger a video of like a forest. Knowing that I couldn't go outside, I was like, “Well, I’m going to run inside.” But to make it less boring, I trigger a video in front of me when I’m running. The way that I detected running was kind of the same. It’s like, if my right ankle is above the level of the floor, then obviously, like I’m kind of like running or if I’m moving my arms in a certain way, then it means I’m running. You create your own interactions, but then you can trigger whatever you want from it.

[00:18:28] KP: Could we also discuss your gaze-controlled interface?

[00:18:31] CG: Sure, yes. This one was interesting, both technically, but also conceptually. So that uses another model, because I didn’t need to detect the entire body. There’s a model just for the face, called a face mesh, that has a lot more points around your face. But the one that I used for that one was the one around iris detection. It is really good and accurate. I mean, at least when I tried it. For me, it was pretty accurate. That was really interesting to work with. Because then, you realize that in terms of input, you don’t need that much to create an interaction. If you detect that you’re looking left or right, then that’s only two types of input, but you can use that to create an interface. It is inspired by a project that I saw on the Google experiments. But as it was an Android app, and I was thinking, “Well, I think that with TensorFlow in the browser. I can kind of do the same.” I use the face mesh model with iris detection, and I created an interface where by moving my eyes left and right, I could select different letters and comments that I can send to VS code to write code.

It was an interesting thing because we’re so used to typing on the keyboard and the first thing that we have is like the 26 letters that I actually realized there is – it’s not the great way to start. You shouldn’t start with the keyboard and write every single letter of a word. It would be very slow. But if you write your interface in the way that you can create snippets, instead of writing each letter, then you go actually much faster. The point was to try to create a React component. And yeah, you can kind of do that in a very simple React component that just prints two letters. But you can do that in about 30 seconds with your eyes. It is very much of a prototype. It’s not meant for production, but it was an exploration of what it could look like to write code with your eyes. Then you can expand and add more things if you want. I think it could potentially be sometimes like a production thing, but it would need a lot more time than the time that I have on the side.

[00:20:44] KP: I mean, the very promising part from my point of view is that as a side project, you made a practical technology demonstration. Maybe it's not ready for a special needs person to start using, but you've shown the path and an organization could give a grant or something like that. Do you think there's a viable path to this becoming an assistive technology?

[00:21:04] CG: I would love for that to be something that people can actually use. I think, to me, the point of my experiment is more to show that it is possible. Then knowing that it is, it is about refining it and polishing it. But what I really like is that, knowing that – if I can do it, anybody can do it. Considering the amount of JavaScript developers out there, showing that you can do this with the skills that you already have on the job that you're doing, I would hope that it shows that people can build things like this. I don't know if maybe people are already doing things like this and I'm not really aware. But I've had a few people messaging me after watching the talk and being like, "Oh! I have friend who can't really use keyboards the way that other people do." That would be really useful for them. There's definitely people that could use something like this, that it would be really helpful for them. But then, I don't really know why there's not more money into this space so that tools can be built to help people. But yeah, to me, there's definitely an audience. But then, it's out of my control to know why people don't actually spend more time trying to build things like this.

But at least knowing that JavaScript developers can – even just for people that they know, or for family members, or for friends quickly build an interface that can help people around them. I think that's the main message. Then maybe, if more people get more interested in this, then it will end up being something bigger, you can only hope.

[00:22:41] KP: Well, many, many years ago, I became aware that there were a lot of hacker developer projects to kind of reverse engineer the Microsoft Connect device. I know it's sort of apples to oranges in comparison, but at least at the time, it was sort of like a project for a grad student. There was a barrier to entry on that. I mean, it seems like we've gone from grad student project to maybe side project that the technology has gotten more accessible. Do you think we've lowered the bar or is there still a distance to go?

[00:23:10] CG: I think that things are more accessible now. I think there's still, I'm not quite sure why people don't really get into this. I think there's also – time is limited. I think there's always an image that makes it look like it's difficult to do. I'm not saying that it's easy. I've definitely struggled before I was able to build something. But I do think that it's a bit easier to get into than it was. Because now, for TensorFlow.js, for example, you only need the laptop or a phone, which if you're a developer, you probably already have. Whereas a Kinect, you needed to buy the device or it wasn't – maybe there was like a tool in JavaScript at the time that I wanted to play with, but a lot of the times, I think the SDK was maybe like in in Python or Java, which for frontend developers, it means that you have to learn that on top of the rest.

I feel like to me, the fact that TensorFlow.js is available, means that it opens up the door like a lot to people. And then you have tools built on top of TensorFlow.js like ml5.js, which tried to make it even more beginner friendly. The space is there, and I think there's definitely new opportunities for people to build things more than before. But I think the issue is, with the amount of things that need to be learned in frontend development, and you always have new things all of a sudden. There's like WebAssembly, or people are starting to talk about that Rust in the frontend. I feel like it's really difficult for people to know what to focus on, so it's also that. It's maybe people don't see machine learning as something that would be useful to them, so they go and learn WebAssembly or whatever, new JavaScript framework is out there because they think it'd be more useful for them. So yeah, I don't really know. To me, I feel like it's nice that now it's accessible to more people, but I understand that from the outside, it might still look a bit scary.

[00:25:02] KP: Are there any particular trends, or projects or languages along those lines that you're excited about following currently?

[00:25:10] CG: At some point, I started looking into WebAssembly, because I wanted to see if maybe I could do some Arduino stuff in the browser with WebAssembly. But to me, that's also the moment where it looks like it's quite complicated, so I would need to put a significant amount of time in trying to understand it. But I did read a really good book about WebAssembly, and I tried a few different projects and it was really interesting, I put it on pause for now. I think it's same as everybody. I have a limited amount of time. But that is something that I was quite excited about, especially when you see applications like the two FFmpeg but in the browser, like that's really – applications like this are really interesting.

I've been looking a bit into Rust in the frontend. But again, as this is a total different language, I feel like getting into learning this is a little bit harder. That's why I like when people bring other disciplines into JavaScript, so at least it lowers a barrier for me, so I can learn about things. But that's maybe the two things I've been looking into lately. Otherwise, different pieces of hardware that I want to play with.

[00:26:17] KP: Such as what?

[00:26:18] CG: Well, there's the brain sensor that I have been playing with before, but I wanted to get back to doing more machine learning with row brain data. That's like a good combination of things that I want to learn that I managed to get together in JavaScript. But there was also that some electrodes that I have, where you can also use machine learning with raw data from your body to create different interactions. There's something I wanted to do. I forgot how it's called. It's kind of like a skin interface. It's basically using electrodes that you stick on your arms. You can use raw data and machine learning to not only detect if you're touching a conductive surface, but how you're touching it. It could know if you're touching it with one finger or two fingers based on the changes in frequency. I think it's called frequency swept something, [inaudible 00:27:11] or whatever. But it is interesting, because then you could detect if somebody is touching a table with their elbow or with their hands based on the data that you get, and you can create then interactive desk setups, or at some point, somebody was creating a way to interact with Spotify, just by touching in your own arms. I was like, "I need to do this" and I tried it and it worked, but it was not in JavaScript. So something I really wanted to do was to port that project into JavaScript, but it's been on my list for like three years. I know I want to do it. Now, it's about finding the time.

[00:27:46] KP: Do you have a perspective on Facebook's big branding of the metaverse?

[00:27:51] CG: Ooh! I've seen it everywhere because it's a bit hard to miss. But I haven't really followed what's going on. I haven't really like used Facebook in a while, so I don't really know where they're going. I have seen a bit more stuff on Twitter around AR and VR, with like the spectacles, like the Snapchat glasses. I wanted to get some actually. But it's something that I've always found interesting to kind of like build alternative worlds. But around Facebook, no, I don't really have an opinion. I've kind of like missed whatever happened.

[00:28:26] KP: Well, I just signed up for your creative technology weekly newsletter, tell me a little bit about what I'm going to get my inbox.

[00:28:32] CG: So it's supposed to be weekly, but I've been a bit tired lately. It's maybe like twice a month more. But I started it because I realized that I spend a lot of time curating what's going on in the creative technology space. Then I realized, "Well, I do it for myself. Cool." But as I'm already doing it, what about just turning it in a newsletter for more people to see. Usually, when I have time, it's weekly. Otherwise, it's like bimonthly. So I curate what I think is interesting, the things that I haven't seen before. And usually, there's kind of like a broad range of topics, but it could be machine learning, or hardware, or AR or VR, anything that to me is an interface that is different than just something like Twitter, like a flat interface that you just scroll and type. Nothing wrong with Twitter, I like Twitter. But in the newsletter, I focus more on things that are different that can maybe give some inspiration for developers to motivate them to either try a new technology or see what's out there, the alternative ways of building an interface. And yeah, I think it's just for people who want to know what's possible. You know you don't have to build anything, but I think knowledge is power and knowing what can be done is always interesting. So yeah, I try to gather things that I think are creative with technology.

[00:29:56] KP: Absolutely. Well, Charlie, where can people learn more about your book projects and speaking engagements?

[00:30:01] CG: I tried to update my personal websites regularly. It's charliegerard.dev. This is where I host all my projects when they get to a point that where I think they're shareable, up with them on my site. Otherwise, I usually share things on Twitter @devdevcharlie as well. I usually share everything that I do, again, in the focus of, hey, maybe you didn't know, but you can do that in JavaScript. Yeah, that's usually the two platforms that I try to keep up to date with what I'm doing.

[00:30:33] KP: Very cool. Thank you so much for taking the time to come on Software Engineering Daily.

[00:30:38] CG: Thank you for having me.

[END]