EPISODE 1363

[INTRODUCTION]

[00:00:00] D: Imagine a world where you own some sort of building, whether that's a grocery store, a restaurant, a factory, and you want to know how many people reside in each section of the store, or how long did the average person wait to be seated? Or how long did it take the average factory worker to complete their task?

Currently today, these systems are either not using AI, and instead use a mix of sensors and buttons to track certain actions, or they do use AI, but in a way that's highly specific to their use case, and hard to easily modify for new use cases that come down the line. This is where BrainFrame comes in. BrainFrame is a tool that connects to all your on-prem cameras, and lets you easily leverage AI models and business logic. Alex Thiele is the CTO of Aotu, the company that makes BrainFrame, and he joins me today to talk about brain frame and the vision for a future where computer vision can be run by anyone.
[INTERVIEW]

[00:00:56] D: Alex, thank you so much for coming on the show.

[00:00:59] AT: Thank you so much for having me, David. Glad to be here.

[00:01:02] D: Yeah, of course. So, in your words, what is BrainFrame? And why did you guys decide to build this?

[00:01:10] AT: Yeah, so BrainFrame is a video analytics platform. If you have a lot of existing, like camera hardware, like specifically, IP cameras, internet connected cameras, and you want to somehow start extracting information from it, using AI, BrainFrame is a good platform for you. We initially created it because of a problem that Aotu was having internally, where we were serving a lot of different customers that needed different video analytics problem solved for each other, for themselves. And they kept changing requirements on us, which was causing havoc on our product line, because we didn't know how to best anticipate the next moves.

© 2021 Software Engineering Daily

So, what we did is we created a very generic useful for all camera processing hardware, platform. We made BrainFrame, which can let you get a lot of cameras, and get any type of algorithm. We really separated how these algorithms interact. We can talk about that more, and then you can deploy and you get API's giving you inference results of all the videos, you get SQL database that saves all the data, and you get a nice GUI client so you can configure things.

[00:02:22] D: From what I understand, anybody that is a system integrator, someone from a more IT background, someone who might not be very familiar with computer vision could set this up, correct?

[00:02:35] AT: Yeah, we have kind of two ideal customers. The first is a super technical algorithm developer who doesn't know how to make a video analytics pipeline. And so, they can just deploy on BrainFrame. And then we have the, like, much less technical, but still they know a little bit of SQL, they know how technology works, system integrators. So yeah, system integrators, they can pop open BrainFrame, download it, run it, and kind of configure it with their cameras, without writing a single line of code. And then the idea is, once they have that hooked up, let's say for like a small store, or a restaurant, then they could write a little bit of SQL and start extracting, like, exactly the analytics they want from this store.

[00:03:15] D: If a factory were to go and say, “We're not going to use BrainFrame, we love this new idea of computer vision, though, and we want to figure out how long does our assembly process take, or maybe how many units were completed?” Something along that line of questioning, how long would it take them to build a similar system from the ground up to say, “We're going to look at all of our cameras, and using computer vision, we're going to make and figure out analytics on what they're saying.” If they want to use BrainFrame, how long would it take?

[00:03:46] AT: It's a really good question, especially like factories versus restaurants, because like a restaurant or store, they're just going to, they always have hundreds of cameras. Factories might just have one station for the camera. Let's say you had a factory that had 100 cameras, and they all needed to be connected to the internet, right? So they would want to – first, they have to develop the vision algorithms for let's say, whatever they need to detect or find or extract using AI, then they'd have to build a video pipeline that can actually stream all those

cameras, decode that video, using GPU hardware, for example, so that it runs very efficiently. Run inference on that video in a nice timely manner, real time, of course. They'd probably have to write some API's there to extract that. And they'd have to expose those API's so they can integrate with other factory hardware. Or if you're a restaurant, you'd want to integrate with the point of sale system. Or if you're a store, you'd probably want a dashboard so that the manager can see it. So, you have to write something there using API's to connect technologies.

We estimate that that would probably be, 9 to 18 months of work before you actually get to solving the user's problem.

[00:04:51] D: But taking the API's aside for a second, the actual kind of secret sauce there is the way from, what I understood, is the way you guys process the video, is that correct?

[00:05:06] AT: It's sort of just having everything in a nice big package, right? So, when you deploy on BrainFrame, you just kind of open up the GUI and you start connecting video streams. So, you can have 100 video streams connected or more, right? But you don't have to configure anything. You don't have to think about how those video streams are being processed. You don't have to think about how the algorithms are going to be scheduled and, run as efficiently as possible on whatever hardware you're running on. You don't have to think about saving that data to a database and cleaning up that data over time. It's sort of optimized for this use case platform.

[00:05:40] D: It's very interesting. What I like that you were mentioning earlier, is these algorithms you kind of – what I've seen is, they're talked about as capsules, can you kind of describe a little bit more about the capsules? And kind of what that is really within the BrainFrame system?

[00:06:00] AT: Yeah, we love the term capsules. So, we have a system called the open vision capsules format, which we open sourced under open CV, and we can talk about that more later. But the idea is, how do we standardize each algorithm? So, an algorithm might detect something like might find the bounding boxes for where a person is on a video frame, or a face or a car, then you might have another algorithm that can classify information about these bounding boxes. So, is this car Hyundai? Or is this person wearing a safety hat? Or is this face

wearing glasses? Creating all these different algorithms and picking and choosing what features you want, wasn't really an easy thing to do. But now with BrainFrame, it's literally drag and drop. So, the use case for this capsule system.

[00:06:51] D: So, from what I'm understanding is, you can essentially drag and drop your own computer vision, computer vision pipeline, essentially, by just having these capsules, one capsule, for instance, might detect hats, and also the bounding boxes for a hat. And then another, you know, capsule that you attach to it can detect whether or not that hat is a certain type of hat, and then you can detect, you can add, another capsule to detect another feature on that hat. Is that essentially what is happening? Or am I not understanding that correctly?

[00:07:24] AT: Yeah, that's exactly correct. I think for the hat use case, you might want to go directly to detecting the exact type of hat. But really, the reason you want to separate these things out is because oftentimes compute isn't necessarily your constraining factor. The constraining factor is how are you going to solve this problem for a customer as quickly as possible? So, assembling a data set of bounding boxes of hats, and their specific types might take longer than just classifying a bunch of images of hats into different bins. So, basically, by having this loosey, goosey, drag and drop, connect different data types or different data type system, you can actually shorten the time to production for building a new feature for a customer, a new AI feature.

[00:08:15] D: What's most I think exceptional about this is all of this can be done with a system integrator, one person who has an IT background can simply put all this together.

[00:08:26] AT: Yeah. So, we have like a big library of existing algorithms who got you like people, traffic, like some fun ones, like mask detection, license plates, you can read text on screen. So, there's a lot of features, they can just download it and deploy it and start playing with the analytics from that. But the idea is that you're not locked into this platform, you have this open source format for creating your own algorithms and deploying them. So, we want people to build IP and just deploy it on BrainFrame, because it's the most convenient platform for them. So, a company might say, “Hey, we want to build some secret sauce that can tell you how fast a car is moving from just two frames or something.” So, they could make that capsule and they don't have to make that public, but they can deploy it through BrainFrame.

[00:09:13] D: And that's the question I also wanted to get into, which is, so a lot of these customers that have these problems, if I'm understanding correctly, they're probably not a very technically savvy business. Is that correct? Or is that correct assumption?

[00:09:29] AT: Yeah, I mean, you have the two customers I outlined earlier, you have the super smart, like people who are creating these algorithms. So, those are the best users of the capsule system. And then you have the like less technical system integrators who just want to solve their end customers problem, right? So that the system integrators, they know how to set up all the cameras in the restaurant and how to get like an NVR set up there to record it and maybe set up the point of sale system, and hopefully also know how to set up a BrainFrame instance, right?

[00:10:02] D: Yeah. But what happens for I guess that second tier of customer that we were talking with the restaurants, where they're not very technically savvy, what happens there when they need custom logic, or they have a problem, that isn't something that can be solved with the capsules that are available, and maybe they're not technically savvy enough to essentially write their own capsules?

[00:10:26] AT: Yeah. Typically, what we do is we go like for different verticals. So, if it's a vertical we're interested in working in, we will develop a capsule for that company, or we'll talk to a company that's interested in making an algorithm to solve that problem and then we will do the deployment. So, for example, an example of this happening is we've had a customer who wanted to detect safety vests and safety hats on a factory and construction setting. And so, for them, we thought, “Hey, that's a vertical we're interested in working in.” So, the customer provided video data to us. And this is just like, outdoor public data, just videos, right? We had to figure out, “Okay, how do we extract like, 1.3 terabytes of frames, and just get maybe 500 to 10,000 images labeled, so we can deploy a working safety vest detector.”

[00:11:21] D: Yeah. And when you do do that, when you do extract these images, is there any – from what I remember with video, each image in a video is usually pretty low resolution, does that have any kind of effect here? Or is that not a problem?

[00:11:35] AT: Well, you do want to train on the data you're going to be running on in production, right? So, we want to train on video, the data that was extracted from videos, because in production, we're going to be using that exact data. You get tremendous accuracy gains, just by training on data, specifically from the cameras that you might be deploying on. With that said, I think most machine learning models actually resize images to be even smaller than their original frame size when running inference, because you actually don't need that high resolution. There are some some exceptions to that, for example, license plate reading models require high resolution to read the letters there.

[00:12:14] D: Yeah, that actually makes sense. I kind of want to zoom out here a little bit and talk more about doing this on prem, which actually was one of the things I found to be very unique about this whole solution. You described that a lot of this is not done in the cloud. When talking about that kind of problem right there where you have maybe a restaurant, or you have an institution where essentially the people in that institution are not, you know, they're not very technically savvy. How do you run on-prem solutions there? How do I run maybe a system that isn't built to handle a model that was trained, maybe on five or six GPUs?

[00:13:01] AT: Yeah, that's a good question. So, to expand on that, like a very common case is, we talk to retail stores, who have some hardware on site, because they already have maybe four cameras, and they have like a small computer there, that's just a network video recorder. The common request there is if we can run this on-prem solution on this existing hardware, then there's zero cost, zero hardware costs to deploy on that location, which is very attractive. But usually that hardware isn't built to run all this kind of algorithms. So, the answer is we have to optimize really carefully on this hardware, right? And so BrainFrame does everything it can to process video on either the Nvidia GPU, if there's one or on an Intel iGPU, if if there's an integrated GPU, then when we run inference on those algorithms, we also try to target the various architectures that we can deploy to on the GPU side, just because GPUs are much more effective processing vision algorithms then like CPUs are.

[00:14:04] D: Okay. I also want to zoom in a little bit more about the computer vision process. I know earlier, you spoke about that for some verticals, you guys will go ahead and train set of models to kind of help that vertical, maybe construction sites or restaurants. What does it take to train a model from zero to one? Essentially, what do you guys have to do and kind of what are

the challenges that come with that to basically say, a customer says, “Hey, we have this specific problem. We want to know, are our employees wearing their construction hats?” How do you go about and, and train that?

[00:14:47] AT: Yeah, it's been as much of a learning process for us as it has been for customers. Actually, maybe about four years ago, when we were starting to do this, customers didn't have a very good idea of how AI works. And so, they didn't understand how important the data data collection was. But nowadays we a very clear pipeline. So, what we do is we deploy a minimum viable model. Maybe we train a model on like some open data set kind of adjacent to whatever it is we want to detect. We deploy that on premise, and then we start recording instances, whenever the model spikes and thinks. “Hey, I think there might be a safety hat there.” So, we record all those instances and we end up with like, a terabyte of video. Then we take all that video and run it through different processes to remove any blurry frames, or to try to find frames where the model thought there might have been something there.

Usually, it's completely incorrect. It's like 10% right. But at least you don't have to look at the video yourself. And we try to extract anywhere from 500 to 10,000 images, which isn't very much in the grand scheme of things. But because we're always going to train our models on what's called pre trained models, that is like a model that's already been trained on a big data set. So, I already kind of understand how to look at videos, we can use less data.

And so finally, what we'll do is we'll usually have an iterative process where we get data from customers, label it, train a model, deploy on prem, and then that helps get even better data through the pipeline. So, now we have a better model. Now, it's picking up more instances, more of the type of thing that we want to detect so we can train on that even more.

[00:16:24] D: I actually want to zoom in a little bit on that beginning part, because I want to just understand a little bit more there when you set a minimum viable model, and you said you train a minimum viable model, and then you wait for it to spike for instances of a hat. Say we're just using the hat example here. How do you go about doing that? How do you go about knowing or even setting up a model and saying, “Hey, this model, we wanted to check for hats, and we're just waiting for there to be a clip where the hat is maybe most visible.” Maybe I'm getting that wrong, but how do you kind of set up that minimum viable model?

[00:17:05] AT: Yeah. I mean, it can be wrong, most of the time, as long as it's still like a weak signal towards. Anything is better than having a person scrub through the video, basically, because you have a lot of cameras to work with. And so sometimes we just have scripts that download things from like an image source, like, let's say, Google Images. Other times, we've actually used synthetic data. So, we have an internal tool called Synthol where we generate like, renders of 3D models, we use 3D models, and we render out millions of different camera angles, and camera resolutions and blurs of whatever it is we want to train a model for. So, this is other data won't get you to the production, but it'll definitely get you to like a minimum viable model.

[00:17:50] D: If I'm understanding that correctly, you will train on – you'll take maybe like hours of video. And you'll take a model that was trained either pre trained, or maybe was just trained on a subset of hats, and now you're hoping that, that our video will be good be shortened to like 10 minutes, and maybe nine of those minutes will be incorrect, but one of those minutes will be a set of hats so that the person doesn't have to watch that whole hour video to get a set of hats or hat?

[00:18:22] AT: Yeah, but usually, we're talking about maybe seven days' worth of video from 20 cameras, so many months of video that you want to bring down to maybe just 30 minutes of frames.

[00:18:35] D: Is there no risk there that you maybe lose huge chunks of the maybe the most vital hats or that maybe not getting – essentially, maybe you're not getting like a very clear picture of what a hat looks like, because it's being biased by this very early model. Is that a fear? Is that true problem that can happen? Or is that not?

[00:18:58] AT: It definitely is. Yeah, it definitely is. And that's why that's why you want to be really careful about how you apply that model. I think it depends on how rare the occurrence is for this type of thing, right? So, if you want to detect like people fighting in a subway or something, that doesn't happen often enough that you could even have like a minimum viable model. You're going to have to scrub through that. Maybe you want to use like a police report or something to like try to find times and dates and then work back from there.

But yeah, really depends on how rare that is. Because if wearing safety hats isn't rare, that's fine, right? You can miss half the data. But still, you can start training and get one step closer. We've had instances where we have people human in the loop, there was a like, restaurant use case, they wanted to detect something very specific on the food line, like certain type of food. So, they just had a human in the loop and whenever BrainFrame didn't catch something, someone would click a button, and we'd save that timestamp and get that data pulled later.

[00:18:58] D: So, a lot of the times, you actually have usually a human labeler for this?

[00:20:02] AT: It's actually just someone who worked at that restaurant who knew to look at this like computer screen that was running BrainFrame. And just whenever BrainFrame didn't trigger, they would trigger it for it and we would record that that timestamp.

[00:20:17] D: So, was this person essentially just looking at a screen for like eight hours a day just waiting for BrainFrame to kind of mess up?

[00:20:24] AT: They were doing their job. They were doing their job, like normal and brain frame wasn't integrated into the rest of the systems at the time. They just knew they had to press a button when this certain action occurred. So, they just were told, like, “Hey, can you just press this button whenever this happens in the store?” They're like, “Yeah, sure.”

[00:20:43] D: Interesting. Okay, and then from there, so they press a button when it happens, you guys collected in database, and then you can have the timestamp, and then you're able to take kind of that frame and say, “This is a frame of this thing happening.” And then you put it in the computer vision model?

[00:20:57] AT: Yeah, or even the clip.

[00:20:59] D: Interesting. Okay. Wow, this could be a very computationally intensive or long process, it sounds like. It sounds like it's something that really can vary from use case to use case.

[00:21:13] AT: Yes. And that's always a concern, right? It's all about focusing on how do you minimize that time to model? But also, can you solve these problems without training a model? So, a lot of the time when a customer purchases us with a unique problem, we ask them, “Is there another way to do this with the existing models we already have?” These 40 publicly accessible capsules, and usually that's possible. But sometimes you get these big projects that might let you work in another vertical. So, you take it anyways, because the customer is helping you with real-world data, right?

[00:21:49] D: Yeah. And that's essentially one of the most key things here, the more real-world data you have, the better your model, the better your product.

[00:21:55] AT: Yeah, exactly.

[00:21:57] D: With a lot of these problems that, restaurants shopping centers, factories have, what are the use cases, I guess, not the use cases. What are the hacks? Or what are the solutions they're currently using today? For the example of a restaurant wanting to know how often are people turning over or being seated? Do they have any solutions for that currently? Or is this just data they just don't have? And they have to kind of estimate it whenever they do any kind of business decision?

[00:22:30] AT: Yeah, that's a good question. I mean, you have a lot of people in the loop, that's for sure. For example, like, in the restaurant, you just have people, you have hostesses knocking down names and keeping track of things. Well, there's a lot of software already existing for restaurants. But what comes in useful here is with the camera, you have a platform now, you're you're building on top of it. Cameras are basically the perfect sensor. They see whatever your eyes see. So, anything people are doing right now, a camera might be able to do more consistently. And that becomes valuable when you have a chain of a thousand restaurants, and different managers are trained slightly differently as to how they enforce the statistics that are supposed to be counted for upper management to be able to make decisions, right? So, having cameras there to count and measure these things in a super consistent way across the chain can help the chain make better decisions as a whole.

[00:23:26] D: I don't know if this would be something that could be possible. But for some of these use cases, have you guys ever come across, this being done with very traditional computer vision methods? And for traditional computer vision methods, if you're not familiar, these are essentially algorithms or business logics that will describe or use math to figure out a computer vision problem. So, for instance, the hat problem, there are probably solutions that mathematically try to describe a hat or use a function describe a hat, and will be able to be a sufficient solution. Is that something that you believe, Alex, could be possible for a lot of use cases you're seeing? Or is traditional computer vision, you believe, for the most part kind of dying out, and AI is essentially replacing it?

[00:24:15] AT: Well, one nice thing about traditional CV is it doesn't have biases, right? And data sets have biases, because the data you collect might have more of some type of person, another type of person. You're going to have bias datasets. So, I think you can never say that traditional CV is going to die out. Also, traditional CV is still really pulling its weight and all kinds of other things like spam and whatever else, right? But it really depends. I mean, in the camera world, AI is killing it. It's really good at extracting useful information in a variety of environments.

If you look at like a very common use case that can be solved with both traditional CV and AI is counting how many people cross the virtual line. So, just want to count how many people let's say entered the store, whenever they cross that entry line, you want to count it. You can do that with traditional CV. You can look at the motion of pixels and assume there's a blob there and count people. And if the cameras really well placed, you can do it just as accurately.

But what I always think about is, well, what about the moment the customer says, “Hey, can we not count when an employee walks into the store? I want to count only customers.” At that point, now you can't innovate, right? You can't tell if it's an employee or not. You've been counting the motion in pixels. So, I think there's always a space for AI and I think it's just easier to do a lot of these things with AI nowadays.

For example, there are some industries where you have to be able to explain how an algorithm came to a certain decision and we're not there yet with ai, explainability. So, maybe that's another reason to use traditional CV.

[00:25:57] D: When you describe that problem you just described where the second you want to know, if an employee passed through, is there a way to use both? I mean, could you use traditional CV to kind of draw that line, and then use computer vision to kind of decide whether or not the bounding box of that employee has gone over? Or do you believe there's just no point you just use at that point, you just kind of rip the whole system out and use a computer vision model?

[00:26:27] AT: I think for detecting people, you're better off just using AI these days. But this traditional CV has a lot of uses. We have, for example, just detecting tags. Think like QR codes, but you can tell exact pose and distance of those tags. That's something that traditional CV can do really well, really accurately across a variety of different lighting conditions. So, we have capsules on our website that are purely traditional CV, and they work great.

[00:26:54] D: I find a lot of this very interesting in terms of cost. So, in a lot of articles out there, if you read on the internet, you'll see that the cost of AI is incredibly expensive. I think this is obviously made in part due to the chip shortage we have today. But generally, I was always pretty expensive to train. Is this, for your guys' purposes, is this something that has been a hindrance? Do you train on prem? How do you guys kind of offset a lot of this training costs for you said, you've trained I think, was it 30 models, you said?

[00:27:36] AT: Yeah, around 30.

[00:27:38] D: Thirty?

[00:27:39] AT: Well, we've got our local machines running GPUs. We did the math and we just decided that it was cheaper to just have machines on prem, that could do the training, especially because we were not training models every day. Usually, the data collection processes can take two months, and then you're going to train a model for a couple hours, and maybe do 10 experiments, 20 experiments over the course of a few weeks. So, it just doesn't make sense to pay AWS $6 an hour for a GPU machine, when you can buy a GPU machine and use it for so much more.

[00:28:19] D: And that's what I find interesting, because in traditional, back end computing, or you know, any kind of application development, usually you start on the cloud. And then when the costs become really big, you try to make a move there, do we stay on cloud? Do we start moving to on prem? At least that's kind of how it is in the last few years. But what I've heard, I've heard this from them, multiple people who develop AI, is they actually start on prem, and they're probably going to continue on prem. Is that also what you're thinking or seeing?

[00:28:52] AT: Yes, I think data labeling and the storage of data is best done on the cloud. There's a lot of competing companies right now, who wants to be the holder of your images with all the annotations. It makes sense to do that on the cloud, because you can then send out those tasks and have a workforce of people label all the bounding boxes you want labeled. But on the model training side, I think it's nice having an on prem computer with a bunch of GPUs, because people overestimate how often you're training. If you have a project, a lot of the design work, data labeling is going to take the longest time and then the final part of the project you're deploy. But GPUs are useful for everything. So, having something – if you have a company that's big enough, on premise, makes a lot of sense.

[00:29:43] D: Yeah, absolutely. The first thing when I first heard about this problem that you guys were going after, the first thing that came in my mind was Amazon Go. I think for those who aren't familiar with that, it's a grocery store where you can essentially scan your phone to get in. You just walk up to something on the shelf, you pick up the item and walk out. This all works, of course, because computer vision tracks your movement throughout the store. Do you believe Alex, that the future of computer vision is going to be very similar to Amazon Go? Do you think maybe not necessary just for grocery shopping, but do you think that we're going to basically live in a world where we're very free of a lot of traditional sensors or toll booths or anything that essentially, requires us to encounter friction whenever we purchase something, or whenever we go somewhere? Do you think that can happen in the next 10, 15 years?

[00:30:38] AT: Well, I definitely think that cameras are the perfect sensor and that they have almost any – you can extract almost anything you want from a camera, if you work hard enough at it. The cost is, it's just the perfect sensor, right? If you can see it with your eyes, a camera can tell it, right? The question is whether people are going to work in all these different verticals, to make those cameras into useful sensors for whatever they want to do.

The Amazon Go thing is interesting, because Amazon, you can't convert an existing store into an Amazon Go store, you have to build it from the ground up. Basically, Amazon decided to turn their store into a robot and they put so many sensors on the ceiling there that there's like five cameras per square foot. And they also have depth cameras, they're using infrared and whatever magic they're doing to find when you pick something up.

I do think that cameras will be more useful in the coming future, I guess. The question is how much people want cameras everywhere? Because I think there's going to be pushback from the public to add more applications that use cameras. BrainFrame is targeting places where there are already cameras and trying to just get more information out of those cameras.

[00:31:48] D: Yeah, it seems like there's almost this kind of push and pull. I mean, Amazon Go creates an almost frictionless experience, right? You walk in, you grab, and you literally go, just like in a name. And it reduces an incredible amount of time for grocery shopping. And I'm guessing, imagine if these applications were done for toll booths, imagine sort of stopping at a toll booth, you just keep going, license plate readers will grab your information and there's no need for us to stop. Kind of like you mentioned, yeah, I could see also, a future or a backlash to a lot of this with people saying, “Hey, I don't want to be film that often or I don't want to be have cameras in every single square foot.”

[00:32:34] AT: Yeah, definitely. I was watching minority report the other day, and the guy walked into a store and said, like, he said his name, and I think nobody wants that world. It's just, you don't want to be recognized wherever you go. So, the question is, whether regulation will be the first one to decide, where does the line get drawn there? How much can you use a camera as a sensor? Because there are useful applications for cameras and sensors. The construction site safety is a very obvious use for that or just counting how many people are entering or leaving a store or reducing turnover in restaurants or many places in a factory. But the question is, where is that line going to be drawn? We know that some municipalities have banned face recognition altogether for police use. So, when will that hit the consumer market as well, similar brands? I'm not sure.

[00:32:34] D: That's the question I think we're all wondering is, tech and privacy and kind of the shift of what's going to go where and I think that that's also something that a lot of people share and one of the biggest concerns they have is that they're now always going to be watched. Previously, we were always watched but that had to be reconciled with the fact I think someone had to look through hours and hours and hours of video or somebody would only see this footage if something bad happened. Let's say, you have 200 cameras either you're paying somebody to look at those 200 cameras, or only if there's a shoplifting incident or some kind of car crash do you actually look at those 200 cameras. And it seems kind of now with computer vision there's almost a bit of a paradigm shift, because cameras that are always on can actually provide instant feedback.

One thing that thing is a funny example is when I worked at Home Depot, I would always have to kind of log my hours, but my boss, if he wanted to know I worked at Home Depot actually, right after high school before going to college, and my boss would have to watch hours of video to know when did I come in when did I leave, and whether or not I'm actually coming in at the right hours or leaving at the right hours. But now it seems like almost he could have this camera on in the store and he could just get a report right of everyone coming in and leaving. Do you think this is our new normal or do you believe that there's actually a lot more nuance on this topic?

[00:35:03] AT: That's a really good question. I mean, it all comes back to the thought of cameras as just sensors, right? If a camera is a sensor, then the question is, well, what other sensor would you use to solve that particular problem? So, your boss could also use a different sensor, your phone, right? He could just have an app and whenever you get to work, you click a button, and it just confirms that you've clocked in and make sure your location is correct. If you leave, then it auto clocks you out, right?

The question is, which one are people more comfortable with? The idea of a camera, it's like, “Okay, it's just a huge array of RGB data. If it can tell when you enter or leave, is that the same as having that where you clock in and out?” Either way, these problems, the friction is going to be lessened. But the question is, how much of the role cameras play into it. My opinion is, we designed BrainFrame with this philosophy, which is that the raw data isn't so bad, especially if it's anonymous. So, BrainFrame, it doesn't even record video. There's no way to record video

with BrainFrame. All it is is just a metadata extraction machine for the purpose of amalgamating, like a graph of thousands of hours of footage and being able to see trends and being able to extract useful high-level data about a location. I think that's fine. The question is, if these technologies are used for less savory tasks, that's where it becomes a really big issue. And we have to choose who we work with and what verticals we go into more carefully.

[00:36:33] D: I can see how that can be definitely be something that requires a lot of thoughtfulness. Well, Alex, thank you so much for taking the time to come on our show.

[00:36:45] AT: Thank you for having me. It's been great.

[00:36:47] D: Absolutely. Yeah.

[END]