EPISODE 816

[INTRODUCTION]

[0:00:00.3] JM: Drishti is a company focused on improving manufacturing workflows using computer vision. A manufacturing environment consists of assembly lines. A line is composed of sequential stations along that manufacturing line. At each station in the assembly line, a worker performs operations on that item that is being manufactured. This type of workflow is used for the manufacturing of cars, laptops, stereo equipment and many other technology products.

With Drishti, the manufacturing process is augmented by adding a camera at each station. Camera footage is used to train a machine learning model for each station on the assembly line. That machine learning model is used to ensure the accuracy and performance of each task that is being conducted on the assembly line.

Krish Chaudhury is the CTO at Drishti. From 2005 to 2015, he led image processing and computer vision projects at Google before he joined Flipkart, where he worked on image science and deep learning for another four years. Krish has spent more than 20 years working on image and vision-related problems in addition to his time working on Drishti.

In today's episode, Krish and I discussed the science and application of computer vision, as well as the future of manufacturing technology and the business strategy of the Drishti.

Before we get started, a few updates from Software Engineering Daily land, pod sheets is our open source set of tools for managing podcasts and podcast businesses. We have a new version of Software Daily, our app and ad-free subscription service. We're looking for help with Android engineering, QA, machine learning and much more.

The Find Collabs hackathon has ended and winners will probably be announced by the time this episode airs and we'll be announcing our next hackathon in a few weeks, so stay tuned. If you want updates on all of these various projects in the Software Engineering Daily world, you can check out the show notes for this episode.

[SPONSOR MESSAGE]

[0:02:14.8] JM: Testing a mobile app is not easy. I know this from experience working on the SE Daily mobile application. We have an iOS client and an Android client and we get bug reports all the time from users that are on operating systems that we did not test. People have old iPhones, there are a thousand different versions of Android. With such a fragmented ecosystem, it's easy for a bug to occur in a system that you didn't test.

Bitbar is a platform for mobile app testing. If you've struggled to get the continuous delivery in your mobile application, check out bitbar.com/sedaily and get a free month of mobile app testing. Bitbar tests your app on real devices, no emulators, no virtual environments. Bitbar has real Android and iOS devices and the Bitbar testing tools integrate with Jenkins, Travis CI and other continuous integration tools.

Check out bitbar.com/sedaily and get a free month of unlimited mobile app testing. Bitbar also has an automated test bot, which is great for exploratory testing without writing a single line of code. You have a mobile app that your customers depend on and you need to test the target devices before your application updates rollout.

Go to bitbar.com/sedaily and find out more about Bitbar. You get a free month of mobile application testing on real devices and that's pretty useful. You can get that deal by going to bitbar.com/sedaily, get real testing on real devices, get help from the automated test bots so that you have some exploratory testing without writing any code. Thanks to Bitbar. You can check out bitbar.com/sedaily to get that free month and to support Software Engineering Daily.

[INTERVIEW]

[0:04:24.6] JM: Krish Chaudhury, you are the CTO at Drishti. Welcome to Software Engineering Daily.

[0:04:28.6] KC: Thanks, Jeff.

[0:04:30.1] JM: Describe a modern manufacturing environment.

[0:04:34.3] KC: Basically, in a modern manufacturing environment, the first thing I want to point out is that in this age of automation, really speaking a surprising percentage of manufacturing is still human-driven. In fact, there is a lot of mechanized activity going on there and everything is streamlined. However, the human touch is not eliminated at all.

For instance, there would be a sequence of stations. There is a line which comprises of a set of stations. There may or may not be a conveyor belt moving from one station to another, but basically there is always one unit of work that gets transmitted station after station after station. There is typically three to five, six, sometimes 10. 10 is on the large side, number of stations on the line and the unit passes through all of them and emerges as a finished product at the other end.

However, quite a large number of cases, there is a human being at every station who is performing the works at their station. This is in a nutshell. This is exactly where Drishti comes in, because we actually study the human being. In this age of automation, we are actually making the human more competitive, if that makes sense. I can elaborate on that.

[0:06:10.5] JM: As you are studying the human, you need some method of collecting information on the processes that the humans are taking part in. How do you gather information about the manufacturing environment and its relation to the humans within it?

[0:06:26.6] KC: Basically, we install video cameras on factory floor. Typically, there is one camera per station. I just described the relationship between lines and stations. We installed the video camera, one looking at individual station. These cameras collect the video right now. They beam the video back to the Drishti cloud, which is where our deep learning engine states this deep learning engine.

The basic data that is basically video and it is gathered via basically off-the-shelf cameras. We don't need any special cameras, even relatively cheap video cameras would do. They collect the data and somehow the data right now where our deep learning engine sits in the cloud, the data is beamed back to the cloud. This is our neural networks/deep learning engine is basically the AI brain of Drishti, now works on these videos.

[0:07:35.5] JM: Describe the state of computer vision for video. How sophisticated is it?

[0:07:40.6] KC: It is beginning to get a lot of attention in recent times. However, the state of the art is much less sophisticated compared to say, static image processing. Object recognition in many senses is a solved problem, not so video interpretation. In particular, one of the fairly complicated things we do is recognizing the action, what the human being is doing in the video. This is quite far from solved.

[0:08:18.8] JM: Why is video interpretation so hard?

[0:08:22.7] KC: There are several reasons. The first reason is that it is inherently more complex information. A static image has only two dimensions. Video is three-dimensional, so the temporal element, the third dimension is time. This third dimension, so everything has to be interpreted in this three-dimensional space. That is the number one complication.

For example, there is this famous problem of – if you are trying to – in a static image, if you see a person grabbing a doorknob, is the person opening the door or closing the door? The door is let's say in some middle state. Now you can actually determine that from a video. You cannot make that call based on a single static image. Video has temporal information, which gives a lot of more possibilities, but also makes the – in some sense, it is giving you more information. If you can process it, you will derive more wisdom from the video. On the other hand, just making sense of it is inherently harder.

[0:09:40.3] JM: If you have video feeds throughout your manufacturing environment, what do you need to derive from those video feeds in order to give meaningful enough information to make those workers on the factory floor, or the manufacturing floor more productive?

[0:10:00.4] KC: We derive a few different kinds of information. The one that may be slightly – not necessarily in this order, but I'll start with actions. From the video, we recognize the actions being performed by the worker at the station. That is number one. Now actions are not arbitrary. We know, we have predefined set of actions that are relevant to that particular factory, particular client and we recognize which one or more of those following actions are being performed.

This helps in the following ways; number one, we can emit in-time warnings about missed steps. Remember, I said that we are making the human more competitive in the face of automation, so let's just pretend it's an automotive factory. A human being worker is working there. We can emit a warning like, "Hey, buddy. You forgot to tighten that nut over there."

Imagine if this warning is emitted right after the fact, it's almost trivial to go back, reach back and tighten that knot. On the other hand, if this automobile with the untightened nut goes out into the world, the further from the point of origin that you detect this mistake, the higher the cost is going to be. If it goes out all the way out on the streets, the cost is – it could even be human life. Missed step identification is one of the possibilities.

Then we also detect what is basically known as a cycle. That is to say in a factory, you essentially do the same thing over and over again. We measure the durations of each of these things. From that, we can derive a lot of analytics which allows the planners to make calls, like where is the bottleneck, how to speed up a certain line, why is this line slower than that line? That questions can be answered. Finally, we also provide a search-like facility. You can say, show me the video where so-and-so action was being performed, or show me the video from 17th of July, 2018.

[0:12:40.1] JM: I believe that the hierarchy that you've laid out here is there's a manufacturing line. On that line, there are stations. At each of those stations, there is a number of actions that takes place. Is that a correct interpretation of what you've said?

[0:12:58.0] KC: Yes. We recognized a fixed set of actions per station.

[0:13:04.0] JM: Okay. Now you can have the people within that manufacturing environment specify the steps of the actions that they are taking. They could associate those steps with videos of themselves performing those steps correctly. That sounds like a complex but feasible training process. Can you describe to me the process of training a model for a given action?

[0:13:35.5] KC: Essentially, what we employ are supervised models. We have human beings who watch the video and identify which one of the predefined set of actions that have been performed in the video, and we label each individual frame of the video accordingly. Obviously, they don't go frame by frame. We have a in-house develop – special-purpose software to actually make it a lot faster.

Ultimately, at the end of the day what these people produce is a labelled video, which is fed into our neural networks. We have our proprietary models and these do extremely well. To my knowledge, they beat the state-of-the-art in that. These are proprietary we have applied for patents on all. These neural networks then taking this training data and the model gets trained. Then as for standard machine learning paradigms, we just deploy it out in the wild and it starts recognizing. It starts recognizing actions that it has not seen before. Actions expects to know them. However, it works on new videos.

[0:15:03.6] JM: How consistent is it from manufacturing environment to manufacturing environment? If I'm screwing in a bolt on a tire, for example. Not a tire, a car. Versus screwing in a bolt on a tank. Is that labeled videostream, is it going to be the same in both of those applications, or do you have to create separate models for tank bolts versus car bolts?

[0:15:30.4] KC: No, it's not that fine-grain. However, we are not at the situation yet where we can deal with – have a single model tackle, enter at the world. It's somewhere in the middle. In other words, several quite often factories belonging to the same industry can be clubbed together. The needle is somewhere in the middle. It's not one model for everything, but it's not a separate individual model for every small task.

[0:16:02.7] JM: Take me through the steps of a Drishti integration with a manufacturing environment.

[0:16:08.1] KC: Yes. That is a fairly complex process, which is not just the artificial intelligence, because there are plenty of other things at play here, right? The initial interaction is where our customer success folks meets the client and learns what – learns about the factory, what are the actions to be that are relevant to them, what are the actions they want us to understand, what are the –

In fact, even before that they have to identify what are some of the initial stations that we should work upon. It may be the whole factory, or in some cases, it's a subset of the lines or stations on which they exist. Then camera installation would happen. Then there is a little bit of data collection. Camera installation is followed by data collection, where we just collect the video. We don't do any analysis on them.

Then our in-house labeling team sits with customer success and they also learn how to label them. We have very fast turnaround. The data gets labeled. Then engineers, we have a small bank of models. This is the point where a deep loading engineer would get involved. The deep learning engineer would identify which one from our bank of models is most relevant to this particular scenario and would start training that model.

In extreme cases, the decision might be we need a new model architecture, none of the existing ones fit. We are trying to minimize that as far – Obviously, we don't want too many models to manage. Then we train the neural networks on that particular scenario followed by deployment. Then obviously, there is a little bit of work in the cloud. Another set of activities that has to follow almost every time, the customer would have a certain IT restrictions so we have to comply with those also.

Sometimes the customer has privacy restrictions. We comply with those as well. We have to make sure that our video process the customers firewall and reaches our cloud. We ensure end-to-end encryption, which is why customer's data is totally safe with us. There is no way anybody could steal, even if they listen on to the video, they wouldn't be able to make sense out of it.

[SPONSOR MESSAGE]

[0:19:21.6] JM: As a software engineer, chances are you've crossed paths with MongoDB at some point, whether you're building an app for millions of users, or just figuring out a side business.

As the most popular non-relational database MongoDB is intuitive and incredibly easy for development teams to use. Now with MongoDB Atlas, you can take advantage of MongoDB's flexible document data model as a fully automated cloud service. MongoDB Atlas handles all of the costly database operations and administration tasks that you'd rather not spend time on, like security and high availability and data recovery and monitoring and elastic scaling.

Try MongoDB Atlas today for free, by going to mongodb.com/se to learn more. Go to mongodb.com/se and you can learn more about MongoDB Atlas, as well as support Software Engineering Daily by checking out the new MongoDB Atlas serverless solution for MongoDB. That's mongodb.com/se. Thank you to MongoDB for being a sponsor.

[INTERVIEW CONTINUED]

[0:20:44.6] JM: You mentioned this point in the process where a deep learning engineer is assessing the application type and the training set and trying to find which model that you have built within Drishti would be a good fit for this application. Can you tell me more about how you pair models to given applications?

[0:21:14.4] KC: It is mostly visual similarity. If a human being thinks the situation in which this model was trained, that setting is very similar to the problem at hand. Then if a human being finds them visually similar, the machine would also find it similar, because it's all computer vision. As a result, if it looks similar, it's going to work. It's mostly a matter of visually identifying the closest setting under which that model was trained.

[0:21:52.4] JM: Do you use the cloud only for the training process, or do you also use it for inference?

[0:22:01.4] KC: We are flexible. We could go either way. Right now predominantly, cloud is used for both. However, there is nothing inherently restrictive in the technology to pin us one way or other.

[0:22:18.0] JM: It's not so latency-sensitive in most applications that you would need to put the models in the manufacturing environment.

[0:22:28.2] KC: That is correct. We are open to that. However, that opens up a fairly complex maintenance problem at hand. Think about it if we deploy the model on the age, then we have to have an army of people, because things break down, right? Physical hardware breaks down. We would probably have to maintain an army of people who just are ready to hop on the plane at any time, because if the model breaks down, customer is losing data, which is unacceptable. That way, from a maintenance point of view, cloud makes more sense for a startup.

[0:23:12.2] JM: In data engineering infrastructure like this, you need probably a cueing system, a machine learning framework, a set of streaming systems that you're using. Can you tell me about some of the infrastructure choices you've made?

[0:23:29.1] KC: First, the infrastructure choices in deep learning, that has been evolving with time. When we started, we were mostly on café. There is less of an informed choice, rather than – first of all, Tensorflow was in its infancy at that point when we started. I personally was much more familiar with cafe coming out of Google. Our first models were based on café.

Since then, we have made a journey from cafe to Tensorflow. The latest ones, some of them are in PyTorch. We still have a mixed array models. In fact, one of the slightly curious things that none of us understands exactly how, sometimes the same model architecture trained on Tensorflow seems to be a few percentage points less accurate compared to the corresponding cafe model. Since accuracy is golden in those cases, at certain maintenance cost we have chosen to keep the cafe models alive. That is one set of choices. We are certainly slowly moving everything to Tensorflow environment.

[0:25:02.2] JM: I have heard this issue that – maybe not an issue, but Tensorflow is not necessarily the most performant machine learning framework, but it probably has the best documentation, it has the best ecosystem around it, it has maybe – I don't know, maybe the best tutorials. What are the reasons to migrate to Tensorflow if not performance?

[0:25:26.4] KC: No, it's not really performance. Maintainability is one important reason. There are certain other reasons such as it's much easier to maintain a multi-GPU model on Tensorflow. They're just a more well-defined APIs. It's much easier to take advantage of multi-GPU models on Tensorflow. That in fact has created – that in addition, changing the model is much easier, which is to say its interface with Python is a lot more well-defined. As a result, let's say we want to try several variants of the architecture. It's much easier to do that in Tensorflow compared to café, for instance.

[0:26:23.1] JM: You used a word maintainability. What does maintainability mean with regards to machine learning models?

[0:26:31.1] KC: Maintainability could mean a set of things. Among them, so if going forward, sometimes what happens is my model, there is certain change in the environment with which my model is not working very well. I need to do either incremental training, or I want to retrain. Also, I find bugs – I mean, bugs just happen right at things that don't go according to plan. I'm hesitating to use the word bug in this connection, because this is not a program where – However, machine learning bugs are different. My model was not – is not properly designed or something. Changing anything is a lot easier. This is what we meant by maintainability. Changing the architecture, or the model definition slightly as necessary to tackle an unforeseen problem.

[0:27:37.0] JM: With the Kubeflow project, we've seen a vision for Tensorflow getting better perhaps continuous delivery, maybe things that look like continuous delivery as applied to machine learning, perhaps audit trails.

[0:27:52.4] KC: Yeah. That's how I should have mentioned. The logging is immensely better. This machine is there out in the world and it made some mistakes when I wasn't even awake, let's say. I need to go back and it's much easier to hook up to instrumented for logging, Tensorflow is better at that too.

[0:28:19.9] JM: You spent almost 10 years at Google. You were also at Flipkart for three and a half years. Now you've started your own company, there is a term Google infrastructure for everyone. Now that you've seen a few companies since Google, do we have Google infrastructure for everyone?

[0:28:38.5] KC: No. I still miss the Google infrastructure.

[0:28:44.8] JM: For what reasons?

[0:28:45.5] KC: Various reasons. Honestly in Google, I mostly programmed in C++ and you could do pretty much everything in C++. Many of those things can only be done in Python or cannot be done. Just take, Python MapReduce is very much slower. Performance is one issue. There were libraries for pretty much everything one could want to do inside Google. Here you have to mix and match, what if you – Those set of libraries, something as simple as I want to create a website with charts. There were in-house libraries for doing that. Much of them Google is open sourcing slowly, but not everything. Sometimes the Google versions are a little bit ahead.

[0:29:41.4] JM: What are the big unsolved problems in computer vision for video that are relevant bottlenecks to some fundamental things you would like to do in improving the manufacturing line?

[0:29:56.4] KC: You are specifically talking about manufacturing, may not generally unsolved problems in computer vision of videos.

[0:30:03.0] JM: Let's take it as the second one, more general problems in computer vision for video.

[0:30:09.5] KC: I would say the biggest thing that we – the direction we want to take is look, there is a video going in front of each one of us, right? We make sense of that with that one single machine that nature has given us called the human brain. We don't have a special video for interpreting what's happening on a automobile manufacturer, versus what's happening on a sports arena. No. There is one single machine that is taking care of it all, right? Interpreting all possible videos that can be thrown at us.

We are very far from there. If I train a video to, for let's say surveillance, that same video probably wouldn't work – the same model wouldn't work on manufacturing floor. This generalization of models is a big challenge. We are nowhere close to where we should – where we could be and nowhere close to the human brain either.

[0:31:19.0] JM: Is it important at all to be able to stitch together different videos?

[0:31:25.5] KC: Stitch together on –

[0:31:26.9] JM: I'm sorry, let's say we have two different videos in different areas of an environment. Do we need to be able to stitch together those different videos to do anything, or can we just use the independent videos to get a independently get a macro view of what is going on in a given environment?

[0:31:44.8] KC: Yeah, that seems like an interesting question. As humans, we do that all the time, right? Meaning, if I am looking outside at the moment and seeing – actually, I'm seeing a boat moving across on the water. Now I turn around I see a person walking past my room, I am able to pretty much seamlessly continue to interpret everything. Normal video machine learning models would not be able to do this easily. You would most likely have to – I'm essentially harking back to the generalization point that I made a few minutes ago.

You would need to actually switch models intelligently. You have to know upfront that the previous model was supposed to interpret natural scenes, or some water scene and the other video is suppose – other model is supposed to recognize human activity. In future, one could envision a situation where the model trains on stitched videos and learns when to switch the mode of interpretation.

[0:33:01.2] JM: In human vision, we're doing this constantly in real-time. We have two eyes and our eyes acting individually are able to give us a single field of vision. To what extent do you take inspiration from the human biological system for processing vision?

[0:33:23.3] KC: That particular thing is called stereo vision. Remember though that our two eyes taking two sets of images, but inside the brain they get combined into a single image. This is what the stereo system – there are organisms which don't do that. I think frogs and toads have – I'm trying to remember long-forgotten biology lessons. I think they don't have stereoscopic vision, but humans and higher mammals have this. There is basically in our brain, a single video is playing. It's not two videos don't coexist in our brain. They get fused into one video.

In some sense, the collection of the data may be stereoscopic, but ultimately, we are trying to do what the human brain is going to do, which is to say post the image collection level. When it reaches the human brain, it is already a single video stream. That is what we are trying to mimic with computer vision. We don't specifically pay much – we don't do much with stereo vision. There are systems that work with that, but we are not one of those.

[0:34:42.6] JM: I took a biology class one time where vision was discussed a bit. The way in which your eye and your brain process imagery and turn that into meaning was really complicated and very hard for me to fully understand, even to the extent that we as scientists actually understand it. Is it going to be important for us to actually figure that out end-to-end in order to get a human quality interpretability, or do you think we could take a different path?

[0:35:18.5] KC: Now we are on very subjective grounds.

[0:35:22.1] JM: I completely agree. I completely agree. We'll return to more applied areas shortly. I just wanted to get your take on this.

[0:35:30.2] KC: I will be happy to give you my take. I honestly don't believe there is any need to maintain fidelity with the human vision system. We may take certain interpretations. It's not even very clear whether – I'm not fully convinced that modernly deep learning systems exactly mimic the human brain.

We have found a certain – nobody knows what's exactly happening in a human brain. We have a certain guess. In fact, the layered backpropagation, is there a parallel in the human brain? I don't know at least. My take is let's not over index on that. Instead, what works is the best way to go.

[0:36:15.5] JM: All right. I'm with you. We have glossed over what Drishti actually does to improve the worker's experience. We gave the example of something is moving down an assembly line and a worker tightens a screw and then the thing that they're tightening the screw on moves further down the assembly line and then Drishti alerts the user that, "Oh, you probably should have tightened that a little bit more." What are some other ways that your technology improves the modern manufacturing line?

[0:36:48.8] KC: We also take very detailed time and motion study, like analytics, right? We know how long the cycle lasted, how long each action, what is the duration of each action, so we can generate various analytics on these things, which in turn, enables the factory to basically identify bottlenecks. I will give you a simple example. If you know, let's say there are 10 actions to be performed, we distribute it in five plus five in two stations. Now all these actions are not equal. As a result and this is an assembly line.

Now, let's say the second station has disproportionately heavyweight actions. What will happen is the first station worker would finish and the second station worker would not be finished. Basically, there would be accumulation in between these two stations. This is classically known as an unbalanced line. What that would do – and the entire line moves at the rate of the slowest station on the line. Consequently, unbalanced line is making the entire line slow. Now it is the right corrective action would be to move one of the – or one or more of the actions, so redistribute the actions. We can help with our technology. We can help actually do that rebalancing, but another way.

[0:38:28.0] JM: The CEO and one of your co-founders, Prasad Akella, he built the world's first collaborative robots. You have a background in computer vision. Describe the division of labor between you and Prasad.

[0:38:42.6] KC: Yeah. I feel this is one of the ways industry is a very potent, powerful combination. Prasad is the expert on manufacturing. I understand on the other hand, artificial intelligence and videos. This is why our combination is powerful. It's very hard to find both of these skills in the same person and even a pair of people like that is not so easy to find, apart from the chemistry between us and everything.

Basically, he is the voice of manufacturing. He understands the client and he interprets the business problems. My role is to convert them to bytes. Did that make sense? Basically, we take the probably not fully specified or very loosely specified business definition from there to all the way to the bytes. This is where I come in. My team and everybody, all the entire engineering in

Drishti. We have a very talented team here. This whole team, together we convert that to bytes. The business vision comes from largely from Prasad and his company product management team.

[SPONSOR MESSAGE]

[0:40:20.5] JM: DigitalOcean is a reliable, easy-to-use cloud provider. I've used DigitalOcean for years, whenever I want to get an application off the ground quickly. I've always loved the focus on user experience, the great documentation and the simple user interface. More and more people are finding out about DigitalOcean and realizing that DigitalOcean is perfect for their application workloads.

This year, DigitalOcean is making that even easier with new node types. A $15 flexible droplet that can mix and match different configurations of CPU and RAM to get the perfect amount of resources for your application. There are also CPU-optimized droplets perfect for highly active frontend servers, or CICD workloads.

Running on the cloud can get expensive, which is why DigitalOcean makes it easy to choose the right size instance. The prices on standard instances have gone down too. You can check out all their new deals by going to do.co/sedaily. As a bonus to our listeners, you will get a $100 in credit to use over 60 days. That's a lot of money to experiment with.

You can make a $100 go pretty far on DigitalOcean. You can use the credit for hosting, or infrastructure and that includes load balancers, object storage, DigitalOcean spaces is a great new product that provides object storage, and of course computation. Get your free $100 credit at do.co/sedaily. Thanks to DigitalOcean for being a sponsor.

The co-founder of DigitalOcean Moisey Uretsky was one of the first people I interviewed and his interview was really inspirational for me, so I've always thought of DigitalOcean as a pretty inspirational company. Thank you, DigitalOcean.

[INTERVIEW CONTINUED]

[0:42:28.2] JM: When you were starting the company, did you and Prasad write the first lines of code, or did you just lay out the theory and the vision for the company and then start looking for people to program?

[0:42:41.6] KC: No. I have written code. I have written quite a bit of the origin – I have trained some of the first prototypical models. I wrote in C++ the very first, even the labeling software, the very first version that we wrote. The engineers later threw it away. The very first version was written by me. Even now, I try to code whenever I get some time. It is hard; getting increasingly harder. However, I love to do that. It's almost part of my identity.

[0:43:24.4] JM: I completely believe you and I sympathize with that identity. Are there areas of the code or areas of the problem that you actually feel you understand so much better than anyone else that it is actually, you are a critical – I guess, a critical piece of the programming team? Or is it more like a theoretical understanding that you have and that's your critical role at this point?

[0:43:52.6] KC: I think it's closer to the latter. In other words, so it's not 100% on either camp. It's not 100% in the either end of the spectrum, but I would be closer to – I am pretty much involved in every algorithmic brainstorming that happens. Pretty much most algorithms model architecture, even database architecture, I would be involved in the brainstorming/discussions, giving suggestions. Not all of them are correct. Some of them are stupid. I do constantly give my opinion on things.

That is one way things are shaped. Essentially, that's one thing. The other is prioritization. Sometimes, I try to see the abstractions among things. If you build A and you build B, separately they would fill two things, but I could build C, which pretty much with little modification could serve both as A and B. That abstraction capability is another thing where I try to contribute.

[0:45:18.0] JM: How far are we from a robot that can sweep the floor?

[0:45:22.8] KC: You know about Roomba, right?

[0:45:23.8] JM: I do. Actually, I'm considering buying one right now. Maybe you can advise me on that purchasing decision.

[0:45:31.3] KC: I purchased one many years ago, early 2002 or something. I don't remember the exact date. I was so excited, I didn't really need several hundred dollar broom at the time. I feel this thing needs to be supported. I purchased one many years ago. It worked for me. It sometimes got stuck under the bed, but it worked for me and it was even smart enough to not fall down the staircase.

We are not that far away from that, right? For practical purposes it works. I have heard recent versions are even better, much better actually. I only have experimented with the 20-year-old version, or maybe 15-year-old version.

[0:46:25.6] JM: Well, and as you were watching it operate, could you see the state of machine learning in any of its actions? Did you see any ways in which it was unable to solve problems it should have been able to solve?

[0:46:40.3] KC: Like I said, I haven't used the latest versions. In fact, I am not very sure how much machine learning was in the one that I saw. I wouldn't be able to comment very coherently on Roomba per se. However, I don't really – I would say, if somebody really sat down to create a very nice smart floor sweeper, it should be possible. You should be pretty close, to able to build something close to ideal, just floor sweeper. Remember, we haven't try to do the generalization problem at all. Floor sweeping, it should be doable now. Now if you also want this thing to pick up unwanted objects from the floor, making sure it does not trash dollar bills, but it trashes trash paper, that becomes a harder problem.

[0:47:35.9] JM: Now this is another question, you may not have any insight into. If you want to pass on it, you totally can. Do you have any perspective for how long it'll be till we have a flying version of that? Like a drone that would fly around your house and would vacuum things up? Maybe it's got a voice assistant. Or do we even want flying drones in our house?

[0:47:57.9] KC: Yeah, why not? I took the second question first. It seemed easier. I am not one of those people who feel threatened by technology. Just because something is really different from my existing experience doesn't make it necessarily bad. Flying drones fall in that category rate. They are going to offend my sense of stability in some sense. However, that is not necessarily a bad thing.

[0:48:29.1] JM: Now I can I can imagine a vision in which beyond just making the human on the production line more productive, you start to make robots that work in collaboration with the humans and make the humans even more productive, or create new manufacturing roles altogether, or just eliminate manufacturing roles that are dangerous and only capable of being done by somebody who could just as easily do something else, so you wouldn't be subtracting from the labor force. I can imagine obviously, a path to building robotics. When will you know that you have enough of a company, enough of a set of things that you're doing that you can comfortably start to move into building robots?

[0:49:21.6] KC: Yeah. Boy, that's a very hard question to answer. It's not entirely obvious to me that we even want to totally graduating to change into making robots. Right now, we are betting that human beings are very important to the manufacturing process and we, like I said, are focusing on making the human more competitive. I honestly believe the world is going to be like that for quite some time to come. I'm not fairly sure if we have a very clear path to totally going into automation anytime in foreseeable future.

[0:50:00.0] JM: Tell me a little bit more about the low-hanging fruit applications that you would like to be able to improve in the manufacturing environment? What do you think is up next in store?

[0:50:15.3] KC: Yeah, I can think of multiple. Essentially, trying to predict if the line is going to go as planned, like every manufacturing floor has a plan I am going to produce this many units today, or in the next one week. we can assess if things are progressing according to plan. That is one obvious way to double-click into. Or there are various comparisons. Manufacturing is all about experimenting. We could make those experimentation a lot easier.

[0:50:59.2] JM: If you weren't working on Drishti, what company would you start?

[0:51:02.1] KC: One of the things I did consider was photo automated artificial intelligence-based photo enhancement work. In other words, a lot more automated. You may see this in my Google work. Google photos in the background improves your photo in various ways. Now one could go further along those dimensions. One of the things I actually did for Google was auto-rectification of photos. Essentially, I wish you could see my hands, or the business could see my hands. When you're holding or taking the photo of a certain building or whatever, if your camera is not exactly held in a right front of parallel position, the parallel lines in the scene would actually become non-parallel. This is something we often see. The building seems to – the sides of the building seem to meet at the top.

Now it is mathematically possible to fix this problem. This is just an example, but there are various ways you could enhance a photo and maybe box for the entire ensemble, which doesn't even need a whole lot of interactions for the users. This is one thing I did contemplate before Drishti.

[0:52:31.0] JM: If you were talking to policymakers, and you're free to pass on this question also, if you were talking to policymakers, what advice would you give to them about regulating facial recognition?

[0:52:43.1] KC: Right. This almost touches upon Google Glass. I would in general, tend towards less regulation always. Honestly speaking, if there is a lot of information in the public domain available about you, your face is already in some sense, the look of your face is public property. Everybody can see it. If I recognize and look you up in Google, or additional harming – my doing is not 100%. There is a certain disturbing feeling to it, but I believe you should not knee-jerk, push back on all such efforts.

Instead, more careful case-by-case evaluation is necessary. For example, there is need for privacy. If you're in public space, for instance it may be best to leave things unregulated. That's one way to – I actually have not thought very seriously about this at any point. To me, it seems that excessive regulation in anticipation sometimes does more harm than anything else. I would also favor genetic research, as opposed to clamp down on it for similar reasons.

[0:54:10.1] JM: I know we're up against time. Just a couple more questions. Manufacturing is one of the first areas where we have seen true adoption of augmented reality. I don't know to what extent, but I've seen some really compelling applications of augmented reality. Do you have a vision for how augmented reality fits into your strategy and your vision for Drishti?

[0:54:31.3] KC: By the way, the word Drishti means vision. I don't know if –

[0:54:35.3] JM: Fair enough.

[0:54:36.3] KC: - there is a Sanskrit word. It means vision. It seems somewhat relevant to this question.

[0:54:40.4] JM: Okay. Hilarious.

[0:54:43.0] KC: Augmented reality is very far from practical, still is my view. I have seen – so Google Glass, it was a fantastic out-of-the-world technology, right? You can go anywhere, primarily because it's not a comfortable experience, first of all looking through the corner of your eyes like that. Even in factories, I have often seen augmented reality device people take those things off after a while saying it's – headaches.

The basic technology is no longer – even, should it be completely vision-based like that. I think this technology will go through several more iterations before becomes a reality. It does a lot of very cool things, but when it comes to a point where you almost don't know it's there, that is when it's useful, right? We are very far away from there.

[0:55:48.4] JM: Last question, a very open-ended. I realized, how else will society change due to computer vision that you believe is perhaps notable, or underrated?

[0:55:59.9] KC: To meet a lot of improvement of healthcare is one thing I would like to see for certain. Society would change. For example, I was thinking about this a while ago. With artificial intelligence, it would become a lot more easy to analyze, say MRI scans and stuff. That should reduce human errors by a very significant margin. That is one that – By the way, I should have said this when you asked me what else you would – if not Drishti, what would you have done, this is one thing. I should have said at that point, because I did contemplate that.

Anyway, so these are much lower hanging fruits. In the long run, you can easily have an utopian vision of a world which is run mostly by robots, so that human beings are mostly on the beach having a great fun. That is the ultimate utopian vision, right? Meaning and I am aware of all the downsides to that. Still that's not a bad dream to have.

[0:57:13.4] JM: Krish Chaudhury, thank you for coming on the show. It's been really fun talking to you.

[0:57:17.0] KC: Yeah. Same here, Jeff

[END OF INTERVIEW]

[0:57:21.8] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source, it's free to use and GoCD has all the features that you need for continuous delivery. You can model your deployment pipelines without installing any plugins. You can use the value stream map to visualize your end-to-end workflow. If you use Kubernetes, GoCD is a natural fit to add continuous delivery to your cloud native project.

With GoCD on Kubernetes, you define your build workflow, you let GoCD provision and scale your infrastructure on the fly and GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team, and they have talked in such detail building the product in previous episodes of Software Engineering Daily. ThoughtWorks was very early to the continuous delivery trend and they know about continuous delivery as much as almost anybody in the industry.

It's great to always see continued progress on GoCD with new features, like Kubernetes integrations, so you know that you're investing in a continuous delivery tool that is built for the long-term. You can check it out for yourself at gocd.org/sedaily.

[END]

