EPISODE 1231

[INTRODUCTION]

[00:00:00] JM: Creation Labs is helping bring Europe one step closer to fully autonomous long-haul trucking. They have developed an AI driver assistance system that retrofits any commercial vehicle starting with Volkswagen Crafters and other trucks. Their system uses camera hardware mounted to the vehicle to capture video data that's processed with computer vision to understand the context on the road. This piece of the system was developed by the world's leading experts in computer vision. While the computer interprets what is happening on the road, data is sent to a processing system that can control the vehicle's braking, throttling and steering. The system currently augments a driver's role, but does not replace the need for the driver yet. However, the distance between great drivers and bad drivers is around a 30% difference in fuel efficiency according to Creation Labs. They've trained their systems with data from the best drivers in order to fuel lower costs for vehicles driven by their system. They've also built their system using the highest standards of safety.

Jakub Langr is the CEO of Creation Labs and he has a background in data science. He discusses in today's episode the future of Creation Labs and their impact on trucking and autonomous vehicles.

[INTERVIEW]

[00:01:13] JM: Jakub, welcome to the show.

[00:01:15] JL: Thank you. It's great to be here.

[00:01:18] JM: You work on autonomous driving, specifically autonomous trucking. It's 2021. Give me a refresher. What's the state of the art in autonomous trucking today? Where are we? How far are we from actual deployments?

[00:01:34] JL: Yeah. So obviously there're a number of players with different approaches, right? You can obviously see the sort of Waymos and the Embarks of the world like trying to basically run their first kind of trial engagements. They usually still have a safety driver. I mean, that's kind of the sort of standard operating practice. So it's sort of a little bit disappointing that time went to I think what most people believed. I think one of specificities for us as a startup based in London is that in Europe it's arguably even much worse. There is no real focus on autonomous trucking pretty much anywhere in Europe with basically one or the very most two-two exceptions, which is somewhat disappointing compared to I think that the visions that we've been told.

Obviously on the technical side they're still sort of the Lidar versus no-Lidar debate. I think if you look at it from a sort of less already sort of entrenched view, I think what the correct answer shapes up to be effectively something along the lines of like Lidar is a very useful tool. If I'm being a little bit dismissive, maybe a crutch of sorts that's probably useful in the short term, but probably not necessary in the long term. But there's this inherent conflict between the types of conflict, the types of approaches that existing players have already adopted. And so we're kind of having a really strange debate in my view about what type of sensor to use. So I think that's kind of like sort of 10,000-feet kind of overview of autonomous trucking landscape. Happy to dive into any of those sort of major pillars.

[00:03:21] JM: Can you tell me about the ingest process for all the sensor data and how it gets integrated and turned into a decision?

[00:03:31] JL: Yeah. Effectively, there's obviously a whole, like a large amount of sensor fusion going on depending on what setup you have going on. Our approach has been to basically start with computer vision. I myself as well as my co-founder, Dr. James Hennessey, we both have you know fairly strong backgrounds in computer vision. So we kind of thought like we're going to leverage that to the largest extent possible. I think Tesla has demonstrated that just basically with just some combination of radar and vision you can already have a pretty sophisticated system able to navigate pretty complicated situations. I think that people watching full self-driving beta on the streets of San Francisco can kind of see how far you can get without any complicated sensor fusion and with Lidar and many other modalities. So I think that's about as much as I can say on this topic publicly.

[00:04:35] JM: Can you tell me anything more about like maybe the data pipeline? Just maybe you can speak in general terms or like some of the software infrastructure that you use to process data.

[00:04:47] JL: Right. So I think one of the really important components is obviously having the full context for any particular driving decision. So I think it is no accident that sort of Elon Musk and Andrej Karpathy speak so much about the sort of the 60 data labeling pipeline, which in essence really what they mean by that is a pipeline that effectively understands not just the video data sort of for a particular object, but also is able to track that object across time including its dimensions. So it's sort of kind of not just understands the 2D version of the image, but also implicitly understands the 3D structure behind it. And in some deep level that's probably how humans also understand the world. We have sort of object permanence from very early age and that's something that unless you sort of explicitly enforce in your architecture is probably pretty hard to get as just as an emergent behavior.

[00:05:58] JM: How do you feel about the competitive landscape? Because like there are companies that are really well-financed. They have really large teams working on a problem like autonomous trucking. How do you stay competitive?

[00:06:12] JL: Yeah, great question. So I think one of the core decisions, and I think this might be sort of unique to Europe, is really understanding the regulation side of things, which is sort of the necessary function. And especially in a market where tends to be more on the conservative side, you kind of really need to sort of slowly demonstrate to the regulator that your system is safe. Traditionally, looking at automotive safety cases, the way that they have been done is basically is in almost more deterministic fashion. Meaning you have a certain sensor, let's say radar or Lidar, and I think that's why they've probably been favored early on, because from an automotive safety point of view Lidar gives you very useful guarantees about certain behaviors. I.e., I'm not going to hit something because I have a certain sort of almost first principle physics-based guarantee of a particular sort of distance, and that's incredibly useful.

Sure, Lidar might degrade a lot with weather, but it's still better than sort of just trying to define a statistical safety case for vision systems only. But I think it's inevitable that any time you incorporate a machine learning component, which at the very least for path planning is a very strict requirement, you will inevitably need to sort of have some degree of statistical components in that safety case. So safety case is typically constructed both on a component level as well and on system level. So you basically have to prove to the regulator that your system as a whole is safe and within – Almost no matter what happens or to certain like statistical certainty. Now to have data for that is a pretty long and extensive process, and as a result it makes sense.

To kind of stick to simpler approaches, which is why we started building out very incrementally and really started to be sort of – Started with just a pain point that all logistics pleads have, which is fuel, and really take decisions while under supervision of the human driver where we effectively can start gathering data on statistical safety of our system and sort of build up from there. And, in general, most of Europe has not even had the regulation drawn up for how public roads testing without a safety driver would even look like. In the UK there's some semblance of that, but so far everyone here is tested with a safety driver. So there's sort of an interesting aspects to the way that regulation is done, which is obviously really the biggest sort of real barrier to entry. Like ultimately it's how do you prove these systems are safe. It's probably what's held back this whole industry by so much compared to what was expected.

[00:09:13] JM: Can you explain the regulatory situation in more detail? Like how specifically are regulators handling the maturity level of self-driving today?

[00:09:24] JL: Sure. Essentially, if that's fine, I'll probably stick to the UK, which is the regulatory environment that I know best. And it's interesting because in many ways it is kind of halfway between what U.S. is doing and what the rest of Europe is doing in many ways, because obviously though at states, some of this legislation is done on the state level rather than the federal level. But in the UK there is even an interesting ongoing debate about how should Tesla be sort of allowed to actually run their systems, because even though they're obviously done saying that explicitly, but there's increasing levels of automation available in especially the Tesla vehicles. And so there is a now an ongoing sort of revision of the regulation around sort of automated lane keep assist systems, which is really kind of the first attempts of most OEMs to really get into anything that resembles machine learning.

Tesla obviously autopilot goes far beyond what automated lane keep assist, which is literally just a system that kind of when you can see visible lane markings it will keep you in the middle of the two, which is something that CMU has done in 1990s I believe on a prototype vehicle, which is some sort of very simple contrast-based algorithm. I think it was just some sort of gradient of the contrast at that point. It was very simple. And so that's kind of where the most deployed systems fall.

Now, obviously, there're ambitions to go further. So there is a sort of code of practice for testing of automated and highly automated vehicles that in principle allows level four testing within a defined, what's known as an operational design domain. Basically you define the settings under which your system is expected to perform safely and then you're expected to keep to those boundaries, and in principle that allows for sort of testing without a safety driver. So far, no one's done that yet, but in principle that option exists.

There are some further requirements on how you need to run those. So you need to have appropriate insurance. You need to have sort of appropriate data logging sort of requirements. I think not too different from what California requires. So in many ways it's kind of similar to how California chooses to just define automotive safety.

[00:12:07] JM: Can you say more about how your systems differ from the existing models that are out there, the existing known self-driving models?

[00:12:17] JL: Right. So I think we're going pure computer vision, which I think we're doing for two reasons. One, we think that like over the end, like in the end, you will absolutely have a system that's as robust with computer vision alone. I think no human needs a Lidar. There's no reason why a person would need that. Why any sort of driving system would need that either. At the end of the day, I think Lidar also has many technical challenges. It has a fairly limited range compared to cameras. And especially in the UK or throughout Europe where rain is very much an expected occurrence, unlike in Arizona where you can test your Waymo system and sort of without really consideration to what happens when it rains. Here, you sort of have to put that into consideration. So vision is a very logical step because it can go much further beyond that. It can have a much longer range, which especially for trucking is an important consideration given that to stop really takes a while to stop 44 tons.

And really I think the incremental approach is something that I think to some people it can almost sound like going backward, but even the regulators are still kind of afraid of these sort of machine learning systems being deployed at scale especially for vehicles that sort of have the capacity to basically run through a wall. So there is sort of like another component of that.

And I guess the last thing for us is that we're really trying to rather than going with the sort of fully supervised route where you effectively kind of define every like sort of component of the driving task, we're really trying to basically use self-supervised learning as much as possible. And for the constituent tasks that we've identified so far, that is very much an option. So there're frequent jokes about some players having almost a cone labeling person. And I think that that – Like at the point where you need to define what a cone is, what a random piece of trash in the street is, like to explain to the algorithm whether it matters or not. I think that is already a sort of very – You've already ended up with a system that's way too complex for anyone to really meaningfully manage and deploy. I think in the end it's going to be no less complicated to deploy such a system than to deploy a system that's sort of self-supervised once you can prove statistical safety.

[00:14:55] JM: You've mentioned self-supervised learning I think. Is that the term you used? Can you describe what self-supervised learning is?

[00:15:04] JL: Right. So self-supervised learning is really where there's a way to define the data as effectively the label for itself, right? So in our case, it's basically meaning you can define to sort of the human behavior as what the human actually ended up doing as the correct label, whereas as opposed to a traditional paradigm where you basically kind of have a human annotator define every component, it comes with its own challenges for being able to run the system like that, but unfortunately I cannot go into those right now.

[00:15:47] JM: You are somebody who's deeply familiar with generative adversarial networks. Are GANs at all useful in self-driving?

[00:15:57] JL: Yeah. I think GANs are absolutely useful in a couple of constituent components. One application that I'm still very excited about and I think I just wish I could do a bit more of is effectively generative domain adaptation. So I think absolutely every single company that we know of doing self-driving has at some point trained in simulation. I think one major issue with simulation is the lack of perceptual realism. Obviously we have now sort of attempts at photorealism through sort of render sequences that still take way too long to be practically usable. And really what you want when you're training in simulation is to do exactly what you cannot do in the real world, which is to run that simulation million times a second with the version of an agent and then basically sort of do gradient averaging against all those possibilities.

So the way that we solve this perception realism issue is we use something known as generative domain adaptation, which really sort of borrows from approaches like CycleGAN where we effectively take the sort of simulated domain and the real domain and we basically asked the generative model to translate the simulator into the real. One of the magic things about this approach is that this works even without any sort of paired domain. So you just take a bunch of sort of simulated images and a bunch of real images and the system effectively learns what the correct sort of transformation is.

So if anyone's familiar with the technique called style transfer, you can almost think of it as a learned style transfer where you basically define the sort of synthetic style and realism style and CycleGAN almost learns how to transform one into the other. So that's kind of the high-level. More than happy to go into more of the technical detail of how that works, but hopefully that makes sense.

[00:18:08] JM: I would like to hear you go a little bit deeper into the implementation.

[00:18:12] JL: Sure. Essentially, I think let's just do a super quick refresher what a GAN is. So GAN effectively is a generative adversarial network. So if you break it down, you basically have two networks. One called the generator, one called the discriminator, and basically they're adversarial because they compete against each other. So in the original paper the loss function of the generators is literally just the inverse of the discriminator's loss function. Later on that got revised. But effectively it is two networks where one of them sort of wins and the other one has to lose. So you can almost think about it as like a very adversarial relationship between a student painter and a teacher painter where the teacher is trying to identify what paintings are good, and the student painters trying to produce paintings, which is basically what the generator does, and the discriminator is basically trying to tell what is real and what is fake. So that's the sort of super quick rundown what a GAN is.

Now there's a more advanced architecture that came around in about 2016, 2017 called CycleGAN, which instantly became very popular because of the magic of what you could do with it, where effectively you now have a much more complicated architecture of four neural networks that effectively act as sort of generators between two different domains and discriminators for those respective domains. So what that means is you effectively train one generator to not create an image from scratch but just do basically a rendition of that painting in another style. So you can think about it as like here's a Van Gogh picture and now make it a Da Vinci picture or something like that and sort of the same way the other way around. So you basically have a one painter that does from A to B and then another painter does from B to A. And at every point there's a credit that effectively evaluates the realism of that picture.

And because you can translate it twice, so go A to B and then B to A, you effectively define what's known as a cyclical loss. So you can almost think of it as translating a sentence from

English to French and then from French to English and then measuring the difference. And so that way you effectively get a sort of loss function where even without having sort of relevant labels in the other domain you can sort of define a meaningful mappings in both directions from A to B and B to A. So that constitutes CycleGAN. And effectively now the task just becomes how do I translate synthetic labels or synthetic images to real images and then back again?

[00:21:10] JM: So now that we have a little bit of an understanding of some of the infrastructure and the algorithms that you're working with, tell me about the state of the art for building autonomous driving technology. Like what kinds of cloud services are you using? What kinds of machine learning frameworks you're using? Just tell me about the general stack.

[00:21:33] JL: Yeah. I mean, I think in tech you're kind of always told as an engineer to use boring technology for anything that you're not innovating on. So being a fairly small and agile company, that's exactly what we're doing. So AWS is probably the easiest way to get started. And obviously one of the sort of intelligent things is once you know how the AWS ecosystem works, you tend to use it. So I think their marketing strategy definitely worked there. So that's what we use.

And then on the like framework side, like I think there's this kind of ongoing debate of sorts of between TensorFlow versus PyTorch. We definitely lean heavily towards PyTorch. I think I've personally just – I think I made the switch about three years ago and I just personally found it much easier to work with on almost every level. So that's kind of on the high-level stack. And then obviously once you start putting this into embedded systems, you have to do a bit of transformations through like standard things like Onyx, which is a framework for basically defining I think interoperable or interportable kind of networks that can be basically put on a much less powerful compute than you'd find on a massive box in the cloud because that's obviously convenient for training and training time. You can have as much computers you want. But at inference time, you really want to make sure that it's fairly limited in terms of the number of operations that needs to run. So that's also a design consideration when you're running the neural networks. And I think a slight tangent, I think one of the reasons why Tesla has an interesting approach with the sort of self – Designing their own self-driving chip, and I think in principle I think for a lot of other players, NVIDIA has such a strong grasp on the market because, really, once you cross certain threshold your options are either making your own or using NVIDIA. And obviously most people just choose to go the simple thing and use NVIDIA. Tesla designed their own. And I think that that is still uh an interestingly sort of underserved market in terms of like just the sheer chip dominance with just a handful of global players.

[00:24:05] JM: Could you go into the dynamics between TensorFlow and PyTorch a little bit more? You said you prefer PyTorch at this point, right?

[00:24:15] JL: Yeah. I mean, I think when TensorFlow came out, was it 2015? 2016 maybe? Something like that. I think I sort of started trying to work with it and it became very apparent that this was kind of a system that was really designed for internal use of Google. I mean, obviously Google ships TensorFlow, but doesn't ship all the support framework. I mean, they've open sourced more of it over the years, but initially it was very like sort of standalone and sort of kind of try your own luck. And I think when you look into the sort of design considerations that TensorFlow has built out, I think it's been really a system that has been optimized for production from day one. But I don't know if in hindsight that was the right call because the sort of choices, the design choices that TensorFlow has made, always made it sort of run relatively well at Google scale, but not so well for anyone else and especially not for people just getting familiar with the framework.

I remember the error messages that you got in the beginning were completely unintelligible. I mean, obviously to Google's credit, they absolutely got miles, miles better. But I just think that attitude of sort of optimizing for the sort of almost internal Google production use case has never really gone away, whereas PyTorch I think has been sort of making I think intelligent decisions around like we still kind of want – I don't think it's unfair to say. I think it almost underprioritized production use in the early days, but it made it very user friendly.

One of the things that I think improves everyone's productivity and then it's sort of almost the taboo of some machine learning, but it's like just being able to use debuggers well. It's such a simple thing in most of software engineering. But in machine learning, really, there're so many people that either don't even know the debuggers exist or are very sort of hesitant to use them. I think it's such a speed up to anyone's productivity. And you almost cannot use it with any debugger with TensorFlow because it’s sort of – Especially, historically, it had the sort of awkward sort of graph definition step and then execution step. And then if you wanted to see the interactions, you had to keep re-running the program to understand what the problem was, whereas with PyTorch you can put in a breakpoint anywhere and just jump straight to that point, which is always such a massive improvement. I think that really sort of won me over.

And then I think over time, PyTorch, again, they started converging closer together, because PyTorch started building more towards production. They introduced things like Torch Script that allow you to basically pre-compile parts of the graph and make it more performant and make it sort of easier to use on other chips and so on. So there's definitely been an interesting degree of convergence between the two, but I think PyTorch is kind of like is easier to pick up to this day and easier to debug. And I think, really, if we all like put ourselves to sort of human level, like that always speeds up your rate of iteration, and that's really what matters.

[00:27:49] JM: Do you have a testing loop for how you test and gather the data on self-driving trucks?

[00:27:58] JL: Yeah. So we're in the process of sort of building a more sophisticated version of that. But of course having a testing loop is incredibly important. I think there's number of things you can do. Obviously simulation is a very useful component of it because not only gives you ability to obviously train new algorithms and explore a little bit more sort of ambitious variations than you might do in the real world, but also it allows you to check that your performance still holds across certain scenarios. One of the things people frequently talk about is edge cases, and you can sort of test that your algorithm runs against well-defined edge cases. There's even a whole set of industry standards on defining the position of certain objects, the vehicle, the road conditions and so on, which then constitute a safety case sort of – It's not quite a safety case. It's not quite true. But it is essentially a definition of sort of edge case that your algorithm needs to clear.

It comes from sort of safety engineering world where you basically use these kind of like scenarios where your algorithm broke down and where your procedure broke down and then you basically try to make sure that never happens again. So, basically, in many ways that becomes a very important part of your CI/CI. So there's obviously elements you can do like that. I mean, at the same time there's obviously sort of the difference between like software in the loop and hardware in the loop sort of behavior. So you need to make sure that you're really kind of testing against both. And obviously nothing's going to be as scalable as pure software. So you need to be a little bit more intelligent about when you're doing sort of new releases. Like how do you test against the defined pre-existing sort of real-world scenarios? And do you go out and do testing new scenarios with your new model or not depending on how do you make that decision. So I think there's a lot of important pieces that kind of really break down just the traditional software barrier where you really need to think about the sort of real-world behavior, which for a lot of software, you don't really have to think about too much. It's just an extra layer of complication.

[00:30:32] JM: What are some of the problems that you've encountered in productionizing real-world machine learning?

[00:30:39] JL: Yeah, good question. I mean, I think every time I've deployed a machine learning system into production there's always been at least – Even though you think you thought about everything, there's always something that you've missed, which is why people do testing. I'm trying to think if there's like any sort of general kind of rules. I mean, there's all sorts of – Especially when you're defining sort of the model parameters across to sort of correspond to a particular vehicle, there's always a sort of interesting drifts that happened that depending on sort of the weight and the weight distribution of the vehicle, that can cause quite a complication.

I mean, I think in general, like I think at the end of the day it's always really important to have a really rigorous testing process. Sort of starting out with sort of unit testing of the most basic behavior in your software that ensures the safety code is running as it should. All the way up to basically more complicated scenarios. I mean, in simple machine learning models before in my sort of past life, I think the challenges there in principle were always pretty similar, except of course now that the magnitude and the level of impact is much higher. But there're all the standard problems of like model drift, monitoring models in production. It can be quite difficult for some things to ensure that in no edge case has your model performance degraded to some bizarre equilibrium. I think that just takes a lot of kind of thoughtful work.

[00:32:32] JM: So your company is Creation Labs. Is it a company? Is it an experimentation center? What exactly are you doing at Creation Labs?

[00:32:45] JL: Yes. It is absolutely a company. But I think that we're really trying to basically kind of come up with a new approach to autonomy that in some ways ironically it's a little bit more conservative, but we also think we can provide more value faster. I think one of the really interesting components is that you can deliver value to fleets, whether that's sort of fuel and carbon reduction or sort of safety analysis of the sort of machine augmented human drivers. And I think that's one of the things that I think ultimately especially in Europe will probably take the cake, because I think in general regulators like to see kind of very careful and disciplined deployment. And that's really what we're doing. So we're basically working with logistics companies to provide them basically initially with very, very low levels of automation and sort of increasing that over time. So we're already working on effectively improving on sort of longitudinal control of vehicles. So effectively how fast you're going? How do you do that more fuel efficiently? How do you do that more safely? And that's already something that if you can fit two vehicles, that even the vehicles today rolling off production line frequently have absolutely no intelligence embedded in them as a default. I mean, a lot of the OEMs now start to provide additional add-ons, but those are obviously only available at sort of at purchase decision.

So most of the trucks driving on the road today and for the foreseeable future are fairly dumb trucks, and being able to basically go in and retrofit some layer of intelligence can already provide a lot of value. And we think that incrementally we can take over more and more of the value chain and ultimately sort of start delivering sort of level four systems that can – Just based on the statistical safety that we've proven up with the lower levels of automation are road worthy.

[00:34:59] JM: Can you tell me a little bit more about the integration of hardware and software? How do you get the trucks like driving using the software that you've written and what's the integration stack look like?

[00:35:16] JL: Yeah. So right now we operate with a select number of vehicles for like the sort of full “full level 2 system”. So there we sort of have a defined control model that we basically largely leverage the existing hardware and we basically leverage the EPS and sort of the internal CAN messaging to basically control lateral and longitudinal control. There is a sort of lower level of automation product which is basically just kind of a computer vision based, sort of better cruise control to which only then has sort of one integration point and has a sort of lower safety case requirement, but at the same time effectively manages to sort of just integrate fairly simply into the control systems. So trucks especially have a sort of extra layer of standardization unlike regular vehicles, which is a protocol known as J1939, which effectively sits on top of the CAN protocol and that effectively provides a layer of standardization. It is still somewhat flexible. There are still OEM-specific fields, but there's a great amount of sort of standard behavior. So we use that as a sort of the lower level of automation and then we have a higher level of automated systems for select number of vehicles.

[00:36:49] JM: Do you have a perspective on whether there will be a strong lead by the companies that that build their own hardware and have the stack fully integrated like a Tesla versus some sort of partnership or amalgamation of different companies like some sort of supply chain partnership?

[00:37:10] JL: Yeah, it's a good question. I do think in the personal car space, I think in terms of like mass deployment, and that's part of the reason why we think incrementality makes so much sense and it's kind of in a sense is also what Tesla did, is just the variety of edge cases that you can – Sort of the long tail you can capture in the real world. But, ironically, I think like Tesla is like the lead for like the mass adoption of sort of L4 and maybe eventually L5 self-driving largely not because necessarily that they have been building the cars from scratch, but because they can sort of collect all that data and sort of source new interesting cases on a basically weekly, hourly basis, which is something that is so invaluable when building out systems that have to deal with all sorts of weird scenarios.

I think in principle, like the Waymo approach or to many other players, I think Jaguar I-PACE seems to be a very popular vehicle to integrate with. It's a good car, and obviously the approaches that people have chosen so far have been very restricted. But I think there's a real question, and I think – I mean, obviously people from Waymo and et cetera will disagree, but I think there's a real question about how well this can scale. I think, first and foremost, as a machine learning engineer, I'm worried about the sort of ability of that system to solve cases where everything looks so different. And even for sensors that are sort of less sensitive to the visual side like Lidar, I think you still need – The sort of out-of-distribution behavior is it still needs to be validated. And every time you deploy to a new area, and that could be as little as like the next state, right? Like you're running into real questions about will the system work as well as it did in the previous state. And I think it gets much worse across continents or across countries even between just like Germany and France. It's like very hard to imagine how those two systems will scale just with the approach that Waymo has taken.

[00:39:36] JM: Well, just to wrap up, do you have any other thoughts on the future of self-driving and where we're going?

[00:39:44] JL: Yeah. I mean, I do think that it is still an exciting area. Like innovation is going. And I think the one funny thing that I think sort of applies very, very accurately, I think there's this concept of like the Gartner hype cycle. And I do think that we're definitely past the peak hype and I think we're slowly getting into sort of through the – I think it's like the through of disillusionment or whatever it's called and then there's the plateau of productivity. So I think we're slowly getting there. And I do think that it is still one of the most impactful transformational applications of computer vision and machine learning. Like I think a lot of machine learning and AI has so far been really sort of more as a sort of filtering step of some kind. And I think these are like the first big real production systems that are starting to basically leverage machine learning that make a difference in everyone's lives. So I do think there's a sort of still like much excitement ahead as we sort of build out these systems and kind of bring them to every country around the world.

I think there's an interesting kind of cell set of hurdles yet to overcome. I think we're probably still a good number of years away from sort of full level four self-driving for any type of vehicle. I think maybe there's a non-zero chance that the full self-driving beta that's currently out there will not hit another plateau. I suppose that is possible. My expectation is that more likely than not it will. At which point sort of some areas of U.S. might start seeing level four self-driving in personal cars within this one or two years. I still think it's highly likely that it will hit another plateau. And for everyone else I do think that it will take a lot while longer. And then I think in Europe, it almost doesn't matter who you are. You will need to prove out statistical safety over a long period of time because ultimately the problem with edge cases is that accidents happens very rarely, thankfully, but you still need to show that on this large enough sample that you're performing better than humans or at least comparable. And so there's still a long journey there to demonstrate that in every country on their data on their turf, which I think will mostly be the requirement. But I do think it's an exciting world ahead and I think there's still so much to be done.

[00:42:32] JM: Okay. Well, Jakub, thanks for coming on the show. It's been great talking.

[00:42:35] JL: All right. Thank you for having me.

[END]