EPISODE 85

[INTRODUCTION]

[0:00:10.6] SC: Hello and welcome to another episode of TWiML Talk, the podcast where I interview interesting people doing interesting things in machine learning and artificial intelligence. I'm your host, Sam Charrington.

This week on the podcast we're featuring a series of conversations from the AWS Reinvent conference in Las Vegas. I had a great time at this event, getting caught up on the new machine learning in AI products and services announced by AWS and its partners. If you missed the news coming out of Reinvent and want to know more about what one of the biggest AI platform providers is up to, make sure you check out Monday's show, TWiML talk number 83, a roundtable discussion I held with Dave McCrory and Lawrence Chung.

We cover all of AWS's most important news, including the new SageMaker, DeepLens, Recognition Video, Transcription, Alexa for Business, Greengrass ML Inference and more.

This week we're also running a special listener appreciation contest to celebrate hitting 1 million listens here on the podcast and to thank you all for being so awesome. Tweet us using the #twiml1mil to enter. Every entry gets a fly TWiML 1 Mill sticker plus a chance to win a limited run T-shirt commemorating the occasion., We'll be digging into the magic TWiML swag bag and giving away some other mystery prizes as well. So you definitely don't want to miss this. If you're not on Twitter or you want more ways to enter, visit twimlai.com/twiml1mil for the full rundown.

Before we dive in, I'd like to thank our good friends over at Intel Nervana for the sponsorship of this podcast and our Reinvent Series. One of the big announcement at Reinvent this year was the release of Amazon DeepLens, a fully programmable deep learning enabled wireless video camera designed to help developers learn and experiment with AI both in the cloud and at the edge.

DeepLens is powered by an Intel atom X5 processor which delivers up to 100 gigaflops of processing power to onboard applications. To learn more about DeepLens and other interesting things Intel's been up to in the AI space, check out intelnervana.com.

Okay, this time around we're joined by Kristin Grauman, professor in the department of computer science at UT Austin. Kristin specializes in computer vision and join me leading up to her talk on Learning Where to Look in Video at Reinvent's Deep Learning Summit.

Kristin and I dig into the details of her research and talk including how an embodied video system can internalize the link between how I move and what I see, so as to learn how and where to move and its environment. We discussed various policies for learning to look around actively and how an agent can learn to focus attention on the interesting elements of a scene. This was a really interesting conversation and I'm sure you'll learn a ton from it.

Now, on to the show.

[INTERVIEW]

[0:03:24.3] SC: All right everyone. I'm at AWS Reinvent and I've got the pleasure of being seated here with Kristin Grauman. Kristin is a professor in the department of computer science at UT Austin. Kristin, welcome to this week in machine learning in AI.

[0:03:37.8] KG: Thank you. Thanks for having me.

[0:03:39.7] SC: Absolutely. You are speaking here today at the Deep Learning Summit as part of Reinvent and I'm really interested in learning a little bit more about your talk and what you'll be sharing. But before we do that, why don't we start by having you tell us a little bit about your background and how you got interested in machine learning.

[0:04:00.0] KG: Sure. I'll work backwards right now. As you said, I'm at UT Austin where I'm a faculty member. I've been there for 11 years now and my specialty is in computer vision and machine learning. That's the part of artificial intelligence where you want to make algorithms that can understand images and video.

Before coming to UT Austin about 11 years ago, I did my PhD at MIT. Prior to that I was at Boston College for my undergrad. So I got into AI happily as an undergrad. I actually had the chance to take courses that were relevant including a course in computer vision, and from that got the chance to work with a professor doing some small research project. That really got me excited and got me the chance to get into the research world and then head to grad school where I explored some more.

[0:04:52.7] SC: Did you do computer vision at MIT?

[0:04:55.5] KG: Yeah.

[0:04:56.4] SC: As part of C-cell or another lab there?

[0:04:58.4] KG: Yeah, I was in C-cell. I'm old enough that it was the AI lab before it became C-cell. During my time there, we transitioned from a separate AI lab to C-cell.

[0:05:09.8] SC: Okay. What was your research focus there?

[0:05:13.6] KG: So my PhD work was focused on object recognition. In particular we were developing ways to work with what are called local feature representation. So being able to match objects based on local parts that are repeatable, and the key to my thesis in a nutshell was to show how to perform discriminative learning with these sets of local features. We develop something called the pyramid match kernel that was very effective for fast matching of sets of features to do recognition.

[0:05:47.1] SC: Can you give an example of local feature recognition and where that comes into play?

[0:05:51.7] KG: Sure. Prior to kind of the major advances with CNNs, convolutional neural networks, one representation of choice was to use interest operators to find local points and images that are going to be repeatably detectable across skill changes, lighting changes,

viewpoint changes and then describe the content around each of these local points with some invariant or tolerant representation that's tolerant to changes.

[0:06:20.8] SC: Is this different from like an edge detector or something like that?

[0:06:25.4] KG: Edge detector also can be a point-wise operator. So what's really powerful about these local representations was the repeatability under different viewing conditions. Once we could find points that would be the same points even if you scale the image by two or even if you rotated a camera by 20 degrees or if you changed the lighting in a room. With those kind of features being repeatably detected, you have nice invariants so that you're really robust to changes you experience in the real-world when you see the thing again. That kind of representation got going and these features got going originally from multi-view geometry work where you need to be able to match and do triangulation and reconstruct the 3D scene. Around that time, we're talking back when I was doing my PhD, whit was then found to be quite important in a similar way for recognition, because you want to find the object again even when it's had these changes. The learning challenge came up when — If you want to jump up to categorization, not just finding that same object again, but not bus, this bus, but any bus, or not this car, any car, then you need to do some kind of learning on top of that sort of representation.

[0:07:32.3] SC: As a way to generalize what you've learned from the points to the class of object that you're trying to be able to recognize?

[0:07:37.9] KG: Right.

[0:07:39.5] SC: Okay. It's interesting. You describe an element of that that it's kind of invariant to positional changes and things like that. I guess I'm thinking of this experience I had yesterday. AWS announced this developer toolkit called DeepLens. It's basically a camera on basically a small computer that is self-contained and you can kind of train models in the cloud and push them out to this little computer and do inference at the edge.

They did a workshop where you're detecting a hotdog, basically. Hotdog, not hot dog, Silicon Valley reference. One of the things that was real clear is that it was very intolerant to positional

changes in the hotdog. Basically the hotdog had to fill the whole frame in order for it to be able to recognize it.

Are there elements of the approach that you were describing that you worked on grad school that would — Granted that was a squeeze net model, so it's like a limited — It's a very limited model that was designed to fit on this embedded device. There are elements of that kind of work that are being tied to what folks are doing today with CNNs to try to make them more kind of invariant to those kinds of positional shifts.

[0:08:57.5] KG: Yeah. A couple of things, one, my PhD was back in 2006, so we're talking about things that are not what I'm working on now. But those local features in fact have that kind of invariants more strongly than your vanilla CNN representation will for the whole image.

If you want to just treat object recognition as an image classification problem, that's really a simplification, because when you want to recognize an object, it's not necessarily, as you said, just framed right in the view such that it occupies most of the pixels. If it is image classification, super powerful, including even a squeezed CNN.

If you want to recognize an object that's, first of all, sitting in a room full of clutter then you also have to tackle what's called the detection problem, which means localizing and finding where boundaries of objects are or the very least scanning around to make classification decisions.

For the kind of work we do now, we're actually interested in this very question. If I have an agent that's visually intelligent, it's not enough for it to be handed flashcards and asked to name them. It's a stepping stone and it's a huge one that has grown so much in the last 4, 5 years. But they also need to be able to figure which pictures should this agent be taking? Where does it have to look? What is an object? Even if that's an object I haven't seen — I being the agent, haven't seen before during training. Yeah, I think you can — The kind of demo you described is super powerful, but not all problems are — We have to go even further than image classification on a web photo or a photo that's kind of closely zoomed in.

[0:10:30.7] SC: That's actually a great transition to the topic of your discussion later on today, right? You're talking about — Why don't you tell us a little bit about what you're talking about today?

[0:10:42.1] KG: Sure. My plan today is to give an overview of one segment of my group's work, and what I'm going to focus on is the theme of learning where to look in video. Again, when we think about training of today's state-of-the-art object recognition system such as those that take deep convolutional neural networks, train them on a dataset like image net for which you have a million images, say, with thousand different categories you can name.

Benchmarks like that and training sources like that treat the problem only impart, and this is because they bake in intelligence about how those photos even came to exist. These are human taken photos and their photo is further more — They have the good composition a human photographer would make. Furthermore, they were chosen to be uploaded on the web to even be good enough as an example that someone wants to see.

If you contrast that what you get if you strap a camera to a person's head or if you strap a camera to a robot's head or a vehicle, all such kind of what are called egocentric of first-person perspective views coupled with video, meaning ongoing observation. Not just a well-chosen moment in time, but just continuous video. Then you'll see that the image content and quality is quite different. If you're not going to rely on that baked-in intelligence about human taken photos, then probably your job in the system is to decide where to look in the first place.

My talk today, that's kind of the motivating disparity from going from labeling photos that humans took to having a dynamic camera in the world that captures video on an ongoing way and has intelligence about which parts of it matter or which parts are recognition worthy. That's my theme, and then I'm going to talk about that on a view fronts. One is to look at how to — Systems that learn in an embodied manner. If you think about snapshots on the web as disembodied, because they're just these moments in time that humans took. Well then if you have embodied learning observations, you might be able to do something more.

Take this as a loose inspiration, we certainly know biological systems build up their visual representations not from flashcard learning, like the web photos you could take to be, but

instead from interacting, moving in the world and having the context of that motion and interaction as part of the learning process. Think of a baby doing this, for example, and there's enough evidence on the cognitive science side and that's actually crucial, like a point of study with kittens, a famous one back from the 60s where if you deprive a kitten of the ability to control its own motion, it has severe detriments to visual perception to vomit and even if it sees the same things that a kitten who can control its own motion sees.

[0:13:36.2] SC: Interesting.

[0:13:36.7] KG: That's kind of the first thing you look at, that is motivation. We've been studying how to perform visual learning in an embodied context, and one of our steps in that direction is to take first-person egocentric video. A video — In our case, this one's captured on a vehicle where we don't just see the pixels in the video. We also can pay attention to what we call motor signals. So physical measurements about how the agent is moving in sync with the video that we observe.

[0:14:06.3] SC: Now we're talking about kind of direction orientation of a vehicle in addition to the video that it's capturing.

[0:14:13.0] KG: That's right. Yeah. We look at the GPS coordinates and the heading of the vehicle and sensed from outside of the visual sensors and now we look at them synchronized with the video stream. Then the idea is that this video stream, let it be unlabeled, which means no human has sat down and done some meditation on it. It's just video that's been captured.

But the goal was to — But the system discover the structure linking the two so that it's building its own visual representation that's informed by this embodiment. More specifically, we post it in terms of ego-motion conditioned new view predictions.

[0:14:54.9] SC: Ego-motion conditioned new view prediction. Okay.

[0:14:56.9] KG: Yeah. That's a lot of words at once. What we're saying is that, "Okay. Supposed you are seeing something," you, the agent of course," at a current moment in time. Now, can we

have a representation where it's predictable for that agent how things will look if it moves in a certain way?"

[0:15:13.3] SC: Okay.

[0:15:14.3] KG: You can teach that from unlabeled video if it knows senses its motion, sees what it sees. It's going to learn that connection between how I move and what I see as a function of my motion.

[0:15:24.6] SC: If I can just take a step to kind of paraphrase here, we've seen — There's work that's been done on just taking still video from a single perspective and trying to predict future frames based on what the learning system has seen so far, and what you're doing is you're taking that a step further by coupling. First of all, the learning system isn't static. It's in motion and it's field of view shifts. So you're trying to incorporate that signal into its ability to predict what it's seeing next as well.

[0:16:00.0] KG: Yeah, you can definitely think of it that way. Furthermore, both the dynamic camera and the embodiment or kind of physical motor signal being part of the learning process are distinct. Thirdly, the desire to have this be part of representation learning for better recognition. Once this learning happens, the idea is this will give us — Will learn this embedding that is capable to do view prediction as a function of ego-motion.

Now you give maybe a video, but also even a static photo and that can be embedded in this space where those benefits of visual perception that you arrive at by paying attention to ego-motion are there so that even the static frame, static image representation is stronger. We'll tackle classic recognition task and bump them up, because this kind of so called pre-training from unlabeled video.

[0:16:52.8] SC: You're talking about embedding tier and representations and I usually hear that in the context of word embeddings and things like that and less so in the context of video. Is that common or is that fairly common in the video world as well?

[0:17:06.1] KG: Right. The word embedding here, I just mean as a learned feature space. You come in with your X, which is your image or your video frame or your video sequence and then there's some F of X you want to apply, and F will be the thing you learn which will embed X into a space that is more appropriate for what you're trying to do.

[0:17:22.9] SC: Got it. Okay. That's kind of the challenges you're going after. Where are you in terms of that research?

[0:17:29.6] KG: Yeah. In this part of the work we have some nice results come out. We train this idea with video captured from a vehicle. There's a dataset called Kitty that's widely used for autonomous driving kind of work. We take that video, no labels, learn representation from it and then tackle a number of recognition challenges. I'll take one.

We take a scene categorization task where your job is to name the category among 400 categories that a new image belongs to. Is it a cathedral? Is it a plaza? Is it a courtyard? Is it a hotel room? Etc. What we found just kind in a nutshell is that with this unsupervised pre-training from unlabeled video, the system will have a 30% increase in accuracy compared to what it will get if it's just training in the traditional way, which means those disembodied photos that are labeled.

This is particularly evident when you are low on training data. If you don't have a million exemplars for cathedrals, say, but you say a handful, or any other class that sits in the long tail of objects, then this is especially important to have this kind of free learning from just moving around the world and looking at things.

[0:18:49.8] SC: Where does the labeling come in?

[0:18:52.1] KG: Yeah. Okay. We can learn this representation purely in a non-supervised way and now do any classification we like, train a CNN, train a [inaudible 0:19:00.3] classifier, train a super vector machine. Depending on the capacity of the model required and the amount of label data you have, go from this pre-train representation to tackle it there.

We've explored two ways, or you could treat our video learning as a regularizer for the classification task and do it jointly. Your question, where does label data come in? When we have test-specific that is label, then either we'll use it in this modular way, pre-train, and now train for the supervised task, or jointly where the video is kind of a supplement to the labeled instances you're using to train for the target task.

[0:19:38.1] SC: Okay. Let's look at each of those in series. The first case, you're pre-training on the video data, coming up with your embedding features and things like that. How does that feed into the training of the next model?

[0:19:50.6] KG: That first case, it's very modular and that now just imagine instead of starting X equals pixel vector, you start with X equals our embedding —

[0:20:00.6] SC: Feature vector.

[0:20:01.0] KG: Yeah.

[0:20:01.3] SC: Okay. Got it.

[0:20:01.7] KG: You're in a vector space. So that's just like off-the-shelf. That would have the advantages of being modular. The recognition task you wish to tackle with this feature space, it can arise in the future and the data doesn't have to be sitting together. Whereas if you treat it the second way, this assumes you've got the task data for your task of interest in hand at that very same moment as you learn from the video, and so you jointly train them.

[0:20:28.7] SC: Okay. You've got some results that show performance improvements relative to what specifically? What model did you baseline against?

[0:20:39.5] KG: Always apples to apples. Whatever recognition model classifier would be used for that same recognition task and whichever label data it would receive, we take the exact same for our method. For example, if it's a CNN which we've tested that in the same — We're talking about the same CNN architecture, plus or minus this learning from unlabeled video. Same amount of labels for both. That's the important baseline to say, "If I just did everything the

same way, but now I also have this benefit of watching video and knowing how I moved, then how much better does that make things?"

[0:21:14.0] SC: Okay. Interesting. What else are you covering in your talk today?

[0:21:17.5] KG: Yeah. This is the first thing I'll look at, and then from there we transition, because what I just described was learning from how an agent moves before a recognition task. Then we think, "Okay. Not only do we have to — Not only would we like to benefit from this ego-motion embodiment being in the world when learning, but also when acting or testing."

We've been looking at active recognition. Active recognition is a problem where you are not passively given the data to recognize, and I keep saying you, and I am always talking about the system. The system is given an environment and it has to make choices about what observations to even collect to succeed in the task. Active recognition systems would want to be able to know where to look around in the scene and sequence to decide what the scene is or to recognize an object, or equivalently of robot with active recognition would be able to hold an object and turn it in a sequence of ways such that it rapidly deduces what the subject is.

[0:22:22.3] SC: Either manipulating the embodiment or manipulating the objects itself to essentially get better information about what it's seeing.

[0:22:32.2] KG: Exactly.

[0:22:33.2] SC: How do you go about doing all of that?

[0:22:34.8] KG: Yeah. Right. It actually flows well from the ego-motion based learning. Now you can imagine that if an agent is going to be smart about choosing its motions, one way to get smart about that is for it to be able to predict how things might look if it moved a certain way, because if you can look for it in time or motion that way, then you can predict which motions you could make that would most reduce ambiguity.

I have a current set of posteriors over all the objects I know or all the scenes I know, and then I can envision how things are going to perhaps change if I move in ways one through N and it

doesn't have to be discreet of course. But then which of those N would most reduce the entropy of those posteriors, right? To say, "Okay. Things are starting to converse more. I think this object is getting more clear."

That ability to look ahead is related to what I was describing for the ego-motion condition view prediction, but you're not done with that. So what we explored, first, to tackle this is so called end-to-end approach where we would jointly train modules to do all the important steps of active recognition. What are they? There are three. One is perception. So a way to take the raw sensor input and map it into some internal representation that's useful for the task, key to representation learning.

Two is action selection. So some component that makes that intelligent choice about which motion to make or which manipulation to issue. Then three, evidence fusion, which says, "Okay. This is happening in a loop," and so as these observations come in, how do I aggregate everything I've seen to inform the next round of action selection or if I'm stopping to give my final estimate.

[0:24:21.1] SC: Okay. It sounds to me like is it fair to say that in a way we're trying to build curiosity in the model? Like the way I'm thinking about it, and correct me if I'm off here, and I'm simplifying, but the robot is looking at a scene. It's determining a set of probabilities of what future scenes might look like if it oriented itself in different ways. One strategy would be for it to orient itself in a way that has the lowest probably, like where it's the most unclear about what's going to happen so as to learn the environment, which strikes me as like a curiosity type of motivation.

[0:25:05.2] KG: Yeah. This is a great point you're making, in fact at least to two things. One is that's exactly the right intuition, and in the case of recognition, it's curious for a goal, right? This is the case where there is some task and the agent knows. It's learning to do well. In fact we're going to be learning in a reinforcement learning manner.

[0:25:25.7] SC: I was just going to ask about that.

[0:25:27.4] KG: Yeah. The next — The system could learn in two ways. It could learn it in a greedy myopic way, which says, "I always want the next," what's called the next best view. Which would be roughly let's pick the one that most increases information gain. But you can also train these reinforcement learning systems for some budget of time that says, "Well, I don't need to always make just one next best. I'd like to think about a sequence of motions that will get me to my resolution." Because then every step, maybe the agent can't teleport out of this building to another one, but it can make a sequence of motions that, in aggregate, it expects to have good influence.

[0:26:06.0] SC: That's kind of analogous to tuning you explore-exploit or how short term the agent is?

[0:26:13.1] KG: Right. How short term it is is definitely related. So it's saying if you have a time horizon for decision making, then you can train this big network consisting of all these modules I mentioned and perception and evidence fusion action selection. Could be trained to target that budget. This is all controlled by how you specify that reward function. You could also target it to be more instantaneous, just always greedily making the next best move if you don't have or it doesn't make sense to have a budget to target. We kind of explore both of those.

When you mentioned curiosity, it really rings a bell for me too because where we've gone since then looking at this is to suppose, "Well, what if we have an agent that has to be smart of how to look around not just for this task that I've pre-ordained, not just for image in that classification or whatever it is, but just for a system I'm going to deploy and needs to be intelligent about looking around before that task gets defined." It's kind of an absolute sense, and this is what starts to sound like curiosity, right? Because it needs to be able to jump into a new environment, look around motions, be smart, but not purely motivated by a closed world of decisions that it's going to make. Does that make sense?

[0:27:24.8] SC: Right. I think about that at the foreign of simplicity, like just looking around this room, objects on the table are going to be more interesting than a wall, for example. So I think that's some of the same things that we traditionally use for object detection, like features and edges and color variation and things like that might percolate up as signals for this kind of model. Is that right?

[0:27:47.8] KG: Yeah. Right. What will a system learn if it's asked to be able to look around intelligently without a recognition goal? So our expectation is it will learn to look at the places that are least predictable from everything else around. So your example of an object on the table and the wall, well once the system has seen a part of the wall, a lot of walls are smooth and so there's little need to evaluate many other glimpses on that wall, because with high probability they're going to be similar. You can learn that. Whereas once an agent cleanses a part of a scene that's interesting, which can be more textured, which also means harder to infirm as seeing pixels of, then it will start concentrating some observations there until it becomes clear what by drawing on regularities learned before for other scenes, that it would help you reconstruct those.

[0:28:37.8] SC: It strikes me that one of the differences between kind of the human visual system and the cameras is that we've got focal area and then peripheral, and so it seems like it's more important for us to move our heads around and kind of focus on different things, whereas a robot can just like do a 360 degree scan and capture the pixels of everything. Why do we even need this learning-based curiosity given the differences between cameras and vision?

[0:29:09.5] KG: Yeah, it's a really good point. That's right. Sensing can almost — In some dimensions is more complete for a robot. Like you said, a 360 capture. In fact, that's the kind of data we work with right now. Don't forget that the robot also needs to move in the world. Even if I have omnidirectional observations at a place and space, what I need to know can be around the corner.

Think about if it's not just narrow field of view glimpses, even if you don't restrict it that way, you still have a need to move in the scene. Similar, I think about that case of an agent robot holding an object in which its own manipulator is occluding part of it or part of the object is behind. So even with omnidirectional view, there's content that's invisible. It's behind the object.

[0:29:54.3] SC: Okay. Interesting. Was there another example or scenario that you walk through, that you're planning walk through in your talk?

[0:30:01.3] KG: Oh, yeah. We've talked about kind of ego-motion informing how the agent learns representation. Then we kind of bring that up into active recognition by learning policies for how to intelligently move around to make recognition decisions or just explore in a curious way. The last thing that I look at is instead of kind of how to look around as an agent-centered question, I think about it actually as a human-centered question.

So we're working with 360 video, which quite an exciting media domain, immersive video and connecting to VR, and the way you right now watch a 360 video as a human viewer is a little bit of trial and error, because you can't see what's behind you. So whether you wear a headset, whether you sit at a computer and mouse around on an interface, like on YouTube to view the 360 content, you are in charge of deciding where to look.

It's a little bit of trial and error in the sense that you may have to watch a video a couple of times to really know where the interesting things are. So while 360 video is so appealing even as a consumer for just capturing everything, so I don't have to make decisions at capture time. Well, you're still left with this decision making at viewer time.

So we looked at this with the where to look question in mind, and what we've been developing is a way for learning how to direct — Think of it as automatic video cinematography. Can we learn how to direct a narrow field of view virtual camera within that 360 sphere? So both in terms of its viewpoint angle as well as the zoom so that you could map a 360 video into a normal field of view video that's 2D and flat and plainer —

[0:31:47.3] SC: And interesting.

[0:31:48.3] KG: Yeah, and got the stuff. Right away, that sounds —

[0:31:52.4] SC: There is a temporal aspect of this as well, because you're not — I'm assuming you're trying to smoothly pan around this 360 view as supposed to flash some sequence of interesting things in it.

[0:32:05.2] KG: Yeah, it has to be carving out a video path that has some kind of motion model as well. Your question almost alludes to the two parts of the approach, and one is to figure out

where are the pieces on this sphere that look capture-worthy and then how do I optimize a pass to hit them as well as possible.

[0:32:24.1] SC: Now I'm immediately kind of brought to two thoughts on how you go about this. One is kind of the extension of all these stuff that we spoke about earlier where you're learning features of interest based on the kind of the observations themselves and trying to identify the stuff we talked about previously.

Another would be like invitation learning, put the human in the headset, have a bunch of people look around and kind of trying to learn a model based on what they find interesting. Are you looking at both of those or —

[0:32:54.6] KG: Yeah. We're actually pursuing something distinct, but the kind of the imitation learning would make a lot of sense and it's something to consider and you can just treat it as a supervised problem where if I've seen where humans tend to look or even better if I get to see video editors edit, then I've got good data to train with. Problem is that — As you can imagine, that's going to be hard to build up enough data for potentially. It's expensive on the annotation side.

Our insight was that this actually can be learned from unlabeled video, and that's because people take a lot of video that's not 360, that is normal field of view. Of course, we know it's online. Furthermore, people kind of have selected video that's worth uploading. What we do is have the agent, the learning algorithm look at hundreds of hours of unlabeled video in YouTube of varying content. We'd like this to be content independent to build up a model of what human taken video looks like.

Now you get your 360 content and imagine chopping it up into all these glimpses throughout the viewing sphere and overtime see their space time chunks, then just trying to score these by saying, "How much like this manifold of human taken video are each of these glimpses like?"

[0:34:08.1] SC: Just to interrupt. Is there a lot of 360 video on YouTube?

[0:34:10.8] KG: Yeah.

[0:34:11.5] SC: Really? I have no idea.

[0:34:12.8] KG: I know I wasn't aware either until we started this project, maybe 1-1/2, 2 years ago. There is, and we've started to look at some of the stats. We're on the research side, of course, but we found stats like 360 camera sales are expected to go by 100% every year for the next six years. I hope I got that right. There's huge growth both in the sale and the use of the camera plus the [inaudible 0:34:36.5] online. Yeah, you can download these 360 videos and 4K from Google. That's what we do to get our dataset.

[0:34:45.6] SC: That's how you acquire the datasets, but remind me again what the insight is to the points of interest there.

[0:34:52.2] KG: Yeah. Rather than have kind of an intensive annotated version of training our system where humans teach it where to look explicitly, we let it be implicit and free label because if I have massive collection of unlabeled video, these are all videos that humans took from normal field of view cameras. Then the notion is when you give me a glimpse, and here a glimpse means some narrow field of view carved out from a 360 video. Let it be five seconds long, say. Now if you take out that glimpse and now do some computation to say, "How close is it to this manifold of human taking content?" Based on some, in our case, 3D convolutional features of that glimpse. Is it close to that space or is it really far? If it's close, that means it shares some visual properties. Indeed, in our case, what closeness will mean will be things like farming effects.

[0:35:47.1] SC: So if I understand what you're saying, you've got 3D video but you've also got regular 2D video and you're mapping scenes from the 3D or you're trying to map qualities of the scene from the 3D to video to the 2D video to identify what looks like a human taken video. Is that right?

[0:36:04.9] KG: Yeah, exactly.

[0:36:05.7] SC: Oh, wow! That's interesting. Okay.

[0:36:08.2] KG: That's where the kind of capture-worthy measure comes from, and it comes from an annotated data and it really does pick up on things like framing composition. So if you have a 360 camera just bounding around the world, I mean supposed not unintelligently driven, then a lot of the views are not well-framed, but there are some portion of it that is. So that's one thing that will be learned, kind of the composition, the framing effects. You can potentially also learn content, so the kind of things that it was filming. The blank wall? No. The scene with the people? Probably.

[0:36:44.7] SC: Wow! Really interesting. You mentioned that these three things that you talked about are just kind of one of a bunch of things that you work on in your lab. Can you give us an overview of some of your areas of interest?

[0:36:55.0] KG: Sure. Yeah. The other areas that we work on today, one is looking at fashion. So we've been looking at — For a long time, we've been looking at semantic representations built on what are called attributes. So these properties, like fuzzy, flat, red, metallic, etc. This is a kind of a way to connect visual properties with language.

We've been working on attributes for many years including developing interactive image search techniques that exploit them. I want to find the image that's like this or the image of a shoes, say that's like this, but piontier. That kind of thing. More recently we've been looking at fashion in the attribute space, but now at fully body images of people and understanding things like style and trend forecasting and compatibility between items. This is one project looking at fashion and vision.

[0:37:44.8] SC: Okay.

[0:37:45.1] KG: We touched on kind of the two parts of my work. Really, one is embodied visual perception, which is kind of on this boarder of vision and robotics to do recognition in the world. We touched on 360 video analysis, which we're working in.

The other elements in my group right now, we have some work looking at image and video segmentation, just kind of very core vision type stuff of finding objects and video and images. Finally, how to do things quickly, specifically recognition. We're looking at how to have a system

that can make only observations it needs in the sense of timely recognition. If I have a very deep network, for example, but I can't afford to run the whole thing, can I dynamically choose which portions of it to run for a given new image, or if I have a video, where we talked about how to do this in an embodied, but even if I'm disembodied and I'm just a machine sitting there processing a video. What parts of the video need attention and what features should I extract on each part?

[0:38:46.0] SC: What specifically are — In the simplest case of an image, what specifically are you doing there?

[0:38:52.4] KG: Yeah. What we've been doing most recently, and this is a collaboration with my colleagues at IBM. We've been looking at if you have — Do you know RezNet? This is one architecture that's quite successful. Has the skip connections between layers and blocks.

We have an approach that will use reinforcement learning to come up with a policy that is input condition to decide how to route through that network dynamically so that ideally maybe you'd like to run every single one, but with time pressure you will then decide which to keep and which to drop. That means — Let's see. I mean for a fraction of the block computation will nearly meet the — Even actually match the accuracy of the full network running that whole area.

[0:39:36.2] SC: Wow! That's really cool. Well, I really appreciate you taking the time out to chat with me this morning. Any final words or ways, places to point folks, ways for folks to get in touch with you?

[0:39:49.3] KG: Oh! Sure. Well, thanks of course for having me. It's great to have this discussion. People who are interested in this work can check out our website from my homepage, and we share the papers but also the code and data surrounding all the things we're doing. So we'd be happy to see anyone being able to use them or build on them.

[0:40:06.8] SC: Oh, great. We'll definitely link to that in the show notes so folks would be able to find it easily from there. All right. Kristen, thanks so much. I really appreciate it having you on the show.

[0:40:14.2] KG: Okay, thank you. Nice talking with you.

[END OF INTERVIEW]

[0:40:20.3] SC: All right everyone. That's our show for today. Thanks so much for listening and for your continued feedback and support. For more information on Kristen or any of the topics covered in this episode, head on over to twimlai.com/talk/85.

To follow along with the AWS Reinvent Series, visit twimlai.com/reinvent. To enter our TWiML 1 Mil contest, visit twimlai.com/twiml1mil. Of course, we'd be delighted to hear from you either via a comment on the show notes page or via Twitter to @twimlai or @samcharrington.

Thanks again to Intel Nervana for their sponsorship of this series. To learn more about their role in DeepLens and the other things they've been up to, visit intelnervana.com.

Of course, thanks once again to you for listening, and catch you next time.

[END]