VIJAY JANAPA REDDI: Hi.
In this video, I want to give you a little bit of a sneak preview
into the kind of applications that we will actually be deploying in course 3.
In order to do that, let me start off by reminding you about the ML workflow.
If you recall from course 2, the ML workflow
is all about how you collect your data, how you pre-process that data
so that you can then prepare that data to be ingested by a neural network
that you're designing, which you're going
to train using tools like TensorFlow and then evaluate for both performance
and accuracy.
And when you're happy with it, you use a converter
to actually convert that model into something
that can be deployed in a wide range of devices,
be it a microcontroller or mobile device.
In this particular course, we'll focus a lot
on the deployment aspects of things, which we are yet to cover,
because, in course 1, we focused very much
on helping you understand how to actually design a model
and how to train a model.
We kind of gave you the pre-existing data
sets that you could bootstrap from.
But, in course 2, we took it all the way from collecting your own data
to actually deploying the model or coming close to deploying the model
by actually doing [? tier ?] flight conversions.
In course 3, we're going to spend the entire gamut going all the way
from collecting data to actually deploying
the converted model onto an embedded microcontroller
and actually making inferences.
And the way we're going to do that is by physically deploying the models
onto an embedded microcontroller that comes
coupled with a couple of different sensors
that allow you to have that full experience.
Now this is a kit that I will talk about in more detail in the upcoming videos.
And one of the other things that's interesting is that this kit,
while that's the hardware portion, coupled very elegantly with TensorFlow
Lite for microcontrollers, which is a state-of-the-art inference framework
from Google.
Now TensorFlow Lite Micro runs in a wide variety
of different embedded microcontrollers.
This just happens to be one microcontroller or one embedded system
that we happen to choose.
I'll give you the rationale for this particular setup later on.
But using this setup, we're going to help you cover that ML workflow.
Specifically, we want to help you understand how to build applications
like keyword spotting, which have a lot of core fundamental attributes in terms
of engineering a solid end-to-end pipeline.
Now you might be thinking, hey, yes, I know about keyword spotting.
I've heard about it in course 1.
And hey, in course 2, I did train a small, little with spotting model.
So hey, what more do I need to know?
I'm going to tell you that there is much more than just getting the model right.
I hinted at this in course 2.
But when it comes to actually deploying the model on device,
you really have to think about where is the audio stream coming from.
It's not a file on your computer.
It literally streams in from the input sensor.
And how do you couple that input sensor with your neural network
and make sure all of these things run in a timely fashion?
And there's also the aspect of post processing.
If some of these things seem new or a little complicated,
yes, they should be because they're new concepts really.
So we'll talk about that.
While dealing with one sensor of audio sensor
is interesting, with Visual WakeWords, we

dealt with another sort of sensor which is the visual instance or the camera.
In this particular course, we will actually use a camera module
in order to implement Visual WakeWords.
But hey, that's not it.
We're actually going to focus on taking the audio sensor and the visual sensor
and doing sensor fusion where we can actually
run the two models together so that we can actually
do audio visual recognition.
In other words, how can I get the machine
to respond only when it recognizes that I'm
physically in its space, which is going to make the TinyML device much more intelligent?
And in showing its application, there is a set
of fundamental new things or concepts that we're going to teach you about,
like model multitenancy, which is very important
and rarely talked about in TinyML.
But it's one of the upcoming areas that is something of quite a lot of value.
Finally, we'll expose you to this super awesome and exciting application
called the Magic Wand, where you move the device in thin air.
And you can see in real time the device recognizing what you're actually doing.
For example.
Here, the machine will actually recognize
that the letter is Z. This sort of gesture recognition capability
has many different applications in many different areas,
especially in biomedical applications.
So more on this later on.
I'm very excited to be talking to you about these different applications.
One of the things that we will do is go all the way
from understanding the characteristics of the sensors
and learning how to couple them onto the embedded device, which is then
running the neural network and learning how
to assess whether the neural network that is running on devices
is actually performing well or not.
And that, my friend, is going to give you the end-to-end picture, which,
as I've always maintained, is extremely important to be a very good TinyML engineer.
With that said, I'll see you in the next video
where I actually introduce the TinyML kit.