

# Developing an Introduction to Embedded Machine Learning Course

Ashvin Roharia  
aroharia3@gatech.edu

**Abstract**— The new field of embedded machine learning has enabled microcontrollers to run complicated machine learning models. Embedded devices for machine learning applications can fulfill many tasks in the industry. Even though there is plenty of educational content on embedded systems and machine learning individually, educational content on embedded ML has yet to catch up. The author developed [Intro to Embedded Machine Learning](#) on Udemy to attempt to fill that void by providing fundamentals of embedded systems, machine learning, and Tiny ML. This course concludes with an interactive acoustic event detection project using a microcontroller or the student’s mobile device. By the end of the course, the student will be able to pick their own classifications and audio as well as train and deploy a machine learning model. This is a great way for beginners to be introduced to and gain valuable experience in the field of embedded machine learning.

## 1 INTRODUCTION

In recent years, technological advances in embedded systems have enabled microcontrollers to run complicated machine learning models (Recognize sounds, 2021). Even though there is plenty of educational content on embedded systems and machine learning individually, educational content on embedded ML has yet to catch up. This lack of educational content, especially for those without a background in embedded systems or machine learning, results in a widening gap in access to education (Baturay, 2015). MOOC platforms such as Coursera, Udemy, and EdX have yet to provide a free, beginner-level MOOC on embedded ML.

If no embedded ML courses are added it will be difficult for new learners to pick up embedded ML without first becoming experts in embedded systems and machine learning separately. If an embedded ML course was added that catered

towards beginners instead of experts, learners would be able to understand embedded ML and see its technological significance. It could become a gateway for future embedded ML engineers. They could better understand the possibilities of TinyML on microcontrollers rather than stumbling on this convergence of fields in their machine learning track or embedded systems track (Warden & Situnayake, 2019).

This content development project focuses on developing an Embedded ML MOOC in Udemy consisting of modules, quizzes, and an example project that will ultimately enable a complete beginner to pick up a microcontroller and turn it into an acoustic event detector. For example, an embedded system that can detect when a dog barks.

## **2 RELATED WORK**

Past studies have been conducted to determine the best approach to teach embedded systems and machine learning courses to students. One study involves how to make a machine learning course for those who do not have an affinity towards technology. It was determined that a machine learning course should be made fun and engaging by educating through a tangible, relatable project (Van Der Vlist et al., 2008). It was also noted that project-based learning can combat the high dropout rate that occurs when a course is too high of a level or not interesting to a student (Carr, 2012). This high drop-out rate plagues the embedded ML MOOCs in the field today.

There is a bit of related work that has been done to teach embedded ML to the students. There are only 2 MOOCs in the field that can be easily found. One is a Professional Certificate in TinyML taught by Harvard professors. This is a high-difficulty course that would not appeal to beginners. The second is a \$13 Embedded AI course on Udemy that doesn't have a personalized project-based example. Both of these courses don't cater to the audience the proposed project is targeting which is composed of complete beginners in this field.

## **3 SOLUTION**

The solution here was to create a course consisting of videos, slide decks, quizzes. The first 3 modules of the course were created to be theory-based meaning the students are introduced to the material via slide decks and quizzes. The Embedded ML Project module is project-based meaning the students would

be introduced to the material via videos and assignments. Finally, the introduction and conclusion modules are videos explaining the course and next steps, respectively.

The project-based assignment includes the instructor creating an interesting ML Embedded example from scratch. So not only did it involve creating tutorial videos for the students, the example project itself needs to be created. This was aided by Edge Impulse's example and tutorials of a sound detecting ML model (Recognize sounds, 2021). This is a tangible acoustic event detection example that has been done multiple times in the past (Temko et al, 2009). Edge Impulse aids in creating SVM and KNN models, both of which can be used for acoustic event detection machine learning (Pandya & Ghayvat, 2021). The semiconductor company, Silicon Labs, recently partnered with Edge Impulse so their board was considered a good starting point for the course development (Silicon, 2021). As the course developer progressed through the course, they determined requiring the student to buy a microcontroller was impractical. The course developer shifted focus from a hard-to-get microcontroller to an easily accessible iPhone. The project now only requires an Edge Impulse account and an iPhone to create a unique acoustic event detection embedded ML project.

### **3.1 Course Outline**

1. Introduction
2. Fundamentals of Embedded Systems
  - a. Microcontrollers Lecture
  - b. Microcontrollers Quiz
  - c. Debugging Lecture
  - d. Debugging Quiz
  - e. Threads Lecture
  - f. Threads Quiz
3. Fundamentals of Machine Learning
  - a. ML Basics Lecture
  - b. ML Basics Quiz
  - c. Supervised Learning Lecture
  - d. Supervised Learning Quiz
4. Fundamentals of Embedded Machine Learning
  - a. TinyML Lecture
  - b. TinyML Quiz

5. Embedded ML Final Project
  - a. Tutorial Introduction
  - b. Edge Impulse
  - c. Create Account
  - d. Connect Device
  - e. Connect Device - Phone View
  - f. Collect Data - Phone View
  - g. Observe Data
  - h. Generate Machine Learning Model
  - i. Retrain Model
  - j. Test Model - Phone View
  - k. Next Steps
6. Conclusion
  - a. How to become an Embedded ML Engineer

## **4 METHODOLOGY**

### **4.1 Teaching Methodologies**

The two main teaching styles in this course are theory-based and project-based learning. All other Embedded ML courses include theory-based content, however, the proposed MOOC will include a unique project-based approach. Real-world and hands-on experience are proven more effective than traditional methods when it comes to learning embedded systems (Sababha et al, 2016). Similarly, learning machine learning can be enhanced by learning in a fun tangible experience such as a gaming approach (Rattadilok et al., 2018). Since both embedded systems and machine learning benefit from the same project-based approach, it is safe to assume embedded machine learning can too. It has been proven that entry-level engineers benefit from this type of learning (Sababha et al, 2016).

For these reasons, a significant portion of this course deals with the student creating their own embedded ML project. The course will provide an example project of building an acoustic event detection ML example of a Silicon Labs microcontroller using Edge Impulse software. The student will be able to train their ML model with their sounds and have their embedded system be trained to detect the sounds of their choosing. The choices of what sounds to detect are

endless and will help personalize this field to the student and help MOOC engagement levels.

## 4.2 Content Creation Methodologies

### 4.2.1 Udemy

Udemy was the platform used to host the course. The module quizzes were created and shared using Udemy's tools. The rest of the course material such as videos, slides, and lectures were all hosted on Udemy. Other hosting platforms such as YouTube and Google Drive were not needed as Udemy was able to store quite a bit of content into the cloud.

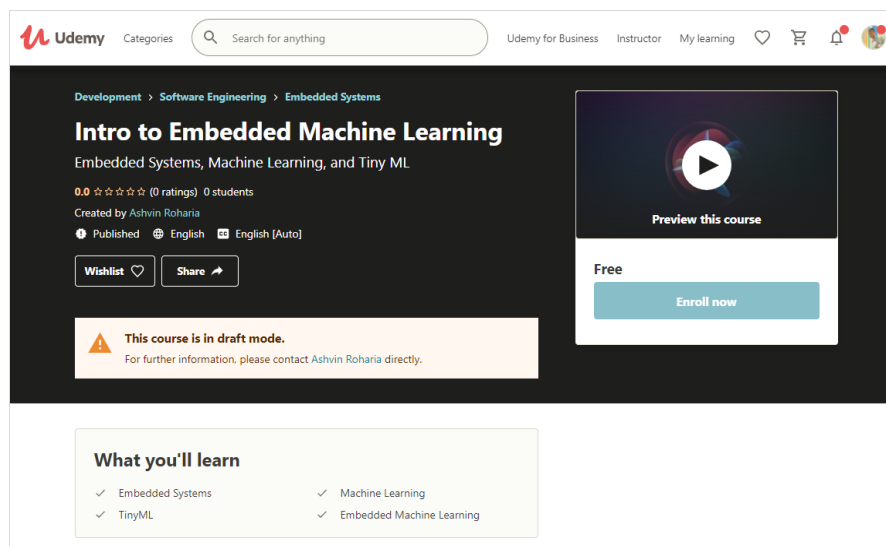


Figure 1—Student view of the developed course on Udemy.

### 4.2.2 Screen Record Capture

The native screen record capture tools on Windows and iPhone were used to record a majority of the videos created. For example, the videos for the Embedded ML Project Module were all screen recordings. The Windows screen recording tool was convenient as it only recorded a certain window instead of the entire screen. It also allowed live commentary to be recorded simultaneously.

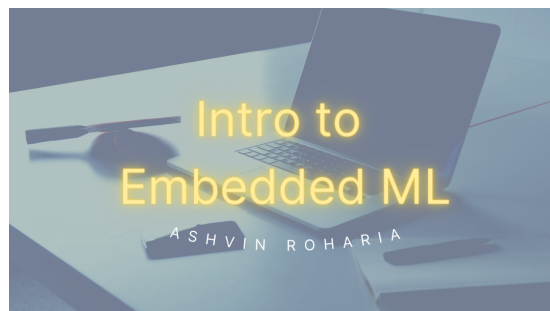
### 4.2.3 PowerPoint

Powerpoint was used to create some presentations for the course such as the Introduction. A normal slide deck was created for that video. The video was also

created within PowerPoint as they include tools to record the presenter's audio for each slide. This was a convenient tool which meant audio could be recorded for each slide rather than the whole slide deck altogether. This allowed for fewer takes as a small slip-up wouldn't require recording the entire presentation again.

#### 4.2.4 Canva

Canva was used to create the course thumbnail image. This is the first image a student would see to Udemy and thus their first impression of the course. A few revisions were made to this image as Udemy rejected the first attempt because they do not permit text in the image.



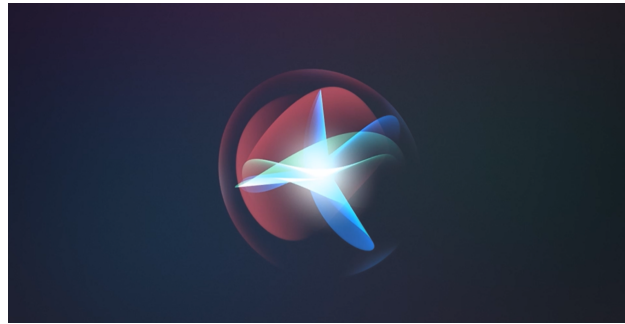
*Figure 2*—First version of course thumbnail. This version was rejected by Udemy.

## 5 RESULTS

The course developer sectioned the final course into 6 modules. The introduction module introduced the student to the field of embedded ML and attempts to build interest in the course. The next 3 modules are the fundamental building blocks of embedded ML - Embedded Systems, Machine Learning, and TinyML. These are all theoretical knowledge-based and include lectures describing relevant topics. The fifth module is the project-based section of the course. This module allows the student to develop their own embedded ML project of their choosing with plenty of guidance from the instructor. The final module is the Conclusion which includes a helpful video for the next steps and what it takes to become an embedded ML engineer. Here is a deeper dive into the content produced for each of the modules.

## 5.1 Module 1: Introduction

This module is used to introduce the student to the instructor and the field of embedded ML. The course developer describes how they have credibility experience in embedded systems since they have worked with IoT products at multiple semiconductor companies. Next, the instructor describes the importance of embedded ML by showing how it is used in everyday life in the form of wake words. A wake word is a phrase such as “Hey Siri” or “Ok Google” that triggers some functionality in a mobile device and uses embedded ML to function.



*Figure 3*—Screenshot from the introduction video. Also used as the course thumbnail. Source: Apple.

## 5.2 Module 2: Fundamentals of Embedded Systems

This module consisted of 3 lectures and 3 quizzes. The topics here all have some connection to the final embedded ML project since covering all embedded systems topics would take too long. Inspiration for the structure of this module comes from the countless embedded systems textbooks in the field today. These textbooks have been teaching tens of thousands of embedded engineers so it was used as guidance.

### 5.2.1 Microcontrollers

The microcontroller section describes what an embedded system is and lists a few embedded design examples using block diagrams. A few topics relevant to embedded ML are highlighted as well. Topics such as microprocessors, CPU, program ROM, and data ROM, the program counter, and the stack. These are all core concepts to embedded systems and well documented in this lecture. The section concludes with two diagrams applying the concepts learned to an ARM-based system. These diagrams show how the microcontroller, microprocessor, RAM, ROM, stack, and program counter are all related to each

other. This lecture's automatically-graded quiz consists of 5 questions ranging from RAM vs ROM, stack and registers usage, and some other topics covered in the lecture. This quiz can be taken multiple times and which makes it less stressful for the student.

### ***5.2.2 Debugging***

This section covers different types of debugging methods when it comes to embedded systems. The instructor believed teaching the students about the difference between nonintrusive, minimally intrusive, and intrusive debugging methods is important for anyone in the embedded field. Example debugging tools were listed for each of these. The short quiz in this section tests the student's ability to critically think by asking a question not listed verbatim in the lecture.

### ***5.2.3 Threads***

Threads and interrupts were explained since they are used in all embedded systems in the field. These topics are relevant to the final example project in the class so it's a useful concept for the student to know going forward within the course. The quiz here requires the students to understand the main thread and interrupt the thread.

## **5.3 Module 3: Fundamentals of Machine Learning**

### ***5.3.1 ML Basics***

The instructor had to do a bit of research before filling out this module. The instructor doesn't work with machine learning daily so needed to research the content online. Since ML is a newer area of study, a lot of easy-to-digest content was found online and used as inspiration. This lecture not only describes the different types of ML but goes into why machine learning is needed. The history of ML is also highlighted to give better context. Giving this bigger picture is important for a student because it allows them to gain a better perspective of what they are learning rather than just jump straight into it. A self-graded quiz is included in this lecture too.

### ***5.3.2 Supervised Learning***

The instructor only describes one type of ML, supervised learning, because this the ML model used in the final project. The instructor considered explaining each



of the types in detail but determined it would be a bit too complicated for the student. Thus, it was deemed as non-effective. Project-based learning would be later incorporated for supervised learning to further enforce the content taught in this lecture. A 4-question quiz completes this section.

#### 5.4 Module 4: Fundamentals of TinyML

The TinyML module is a simple section describing what it is and why it exists. The goal of the course is not to teach programming but to understand what is happening under the hood. This is essentially prep material before the student goes on to implement TinyML in the subsequent module. Since TinyML was found to be universally used in most embedded ML examples, the course developer created a dedicated module for it. This module's quiz is the final quiz in the course thus was made a trickier. This quiz doesn't directly cover the material taught in the course. Instead, it quizzes the student's ability to apply to think outside the box. These questions are based on real-world scenarios and are questions they could encounter in the real world as embedded ML engineers.

#### 5.5 Module 5: Embedded ML Project

The Embedded ML Project module covers the project-based content of the course. The project is meant to be open-ended and allows the student to choose their own audio classes such as fire alarms, baby crying, dog barking, etc. The instructor goes through the entire process an embedded ML engineer would go through - data collection, data labeling, training an ML model, testing an ML model, retraining an ML model, and finally testing the model on real hardware.

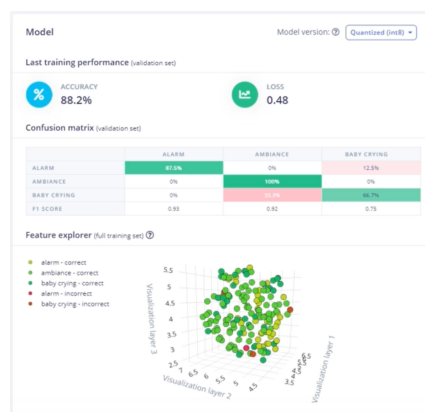


Figure 4—Screenshot from the tutorial in the lecture to create an ML model to classify real-time audio. Source: Edge Impulse.

The student is taken through the whole process in a simple way but is also introduced to common issues such as having a low accuracy ML model. The student is taught how to overcome this issue which promotes independence and pride when achieving a high accuracy ML model.

### **5.6 Module 6: Conclusion**

The conclusion consists of a video explaining what an ML engineer actually does. This is to help teach what is expected of embedded ML engineers and how to become one. By the end of the course, the student will have hopefully gained an interest in embedded ML and enough knowledge to allow them to take their next step, whatever it may be.

## **6 DISCUSSION**

The project progressed well throughout the course. The content creation was made easy via Udemy. Udemy's course creation tools were quite advanced and could host complicated lectures, videos, and quizzes all on their site in a simple way. Simple both for the instructor and the student. Additionally, Udemy provided a lot of creative freedom to the course developer. Other MOOC platforms such as Coursera and EdX would have been more formal as they are geared more towards professional instructors.

The only issues encountered when creating the course were condensing an entire topic into an easy-to-digest module. This is why a new approach was taken during the course which was to focus on the material needed for the final project of the course. This kept the course concise and gave it a bit more structure. This approach also means that this course is not a one-stop-shop for embedded ML. There are definitely higher-level embedded ML online courses that will go into much more technical depth than this course does. The goal of this course is to attract beginners into this field and improve their confidence to go and take those more advanced courses such as the Harvard University one.

## **7 CONCLUSION**

The developer enjoyed creating content and learned more about the subject in doing so. Each of the modules created consisted of multiple hours of researching complicated material to simplify it for a beginner to understand. The course developer believes this was achieved and would be a great resource for future

embedded ML engineers. The hope is to enable students to get interested in the field of embedded ML and give it the attention it deserves.

## **8 FUTURE WORK**

Lengthening the course was a suggestion that was made during a peer review session. This is definitely a consideration that could be applied in the future. There are certainly many more topics that could be discussed. Both within the current modules or even developing additional modules.

Another consideration would be to add more video content. A good chunk of the theoretical portion of the course consisted of lectures in the form of text and images. This is quite a traditional approach that could be seen in textbooks and older classes. Recording videos for each lecture could be ideal and much appreciated for students who benefit from visual learning.

## **9 ACKNOWLEDGEMENTS**

Special thanks to Tyler Bobik and all the Georgia Tech peer reviewers who provided feedback throughout the Spring 2021 Educational Technology course.

## **10 REFERENCES**

1. Baturay, M. H. (2015). An Overview of the World of MOOCs. *Procedia - Social and Behavioral Sciences*, 174, 427–433.
2. Carr, N. (2012). The Crisis in Higher Education. *MIT Technology Review*, 115, 32-36,38-40.
3. Pandya, S., & Ghayvat, H. (2021). Ambient acoustic event assistive framework for identification, detection, and recognition of unknown acoustic events of a residence. *Advanced Engineering Informatics*, 47, 101238.
4. Rattadilok, P., Roadknight, C., & Li, L. (2018, December). Teaching students about machine learning through a gamified approach. 2018 IEEE international conference on teaching, assessment, and learning for engineering (TALE) (pp. 1011-1015). IEEE.
5. Recognize sounds from audio. (2021). Edge Impulse Docs.

6. Sababha, B., Alqudah, Y., Abualbasal, A., & AlQaralleh, E. (2016). Project-Based Learning to Enhance Teaching Embedded Systems. *Eurasia Journal of Mathematics, Science and Technology Education*, 12(9), 2575-2585.
7. Silicon Labs. (2021, January 27). Silicon Labs and Edge Impulse Partner to Accelerate Machine Learning Applications. PR Newswire.
8. Temko, A., Nadeu, C., Macho, D., Malkin, R., Zieger, C., & Omologo, M. (2009). Acoustic event detection and classification. *Computers in the Human Interaction Loop* (pp. 61-73). Springer, London.
9. Van Der Vlist, B., Van De Westelaken, R., Bartneck, C., Hu, J., Ahn, R., Barakova, E., ... & Feijs, L. (2008, June). Teaching machine learning to design students. *International Conference on Technologies for E-Learning and Digital Entertainment* (pp. 206-217). Springer, Berlin, Heidelberg.
10. Warden, P., & Situnayake, D. (2019). *Tinyml: Machine learning with TensorFlow lite on Arduino and ultra-low-power microcontrollers*. O'Reilly Media, Inc.