

## EEP 596 TinyML Course Project - Instructions

### Professor Radha Poovendran

The course project plays a pivotal role in providing hands-on experience in the field of TinyML. In the following sections, we will outline the project timeline, requirements, and potential project ideas to guide and inspire your work. The course project will account for **45% of your final grade**, which is broken down as follows: proposal (5%), final presentation (20%), and final report (20%).

### Timeline

- Beginning of the quarter: Form groups and determine the topic to work on
  - A group can consist of **3-4 members** working together on the same project. Please ensure that the group size does not exceed 4 members.
  - Ideally, we would like to have a unique project for each group.
  - If you have formed a project group with others and determined the topic, remember to enroll your project group (<https://t.ly/PjQ2>) by filling in your group members and selected topic. This should be done before the project proposal deadline.
- Project proposal: due on **May 1st, 11:59 PM PST**
- Final report: June 7th, 11:59 PM PST (Placeholder)
- Project presentation: **June 7th, 6.00PM - 10.00PM PST**. On regular class day and time (Wednesday) during the finals week.

### Project Proposal

- The project proposal should include a TinyML problem that you intend to work on. You should also brainstorm some possible approaches to the problem and write them down in the proposal.
- Make your proposal concise and keep it **within two pages**.
- You should include the following things in your project proposal:
  1. Project title
  2. Names of group members
  3. Dataset(s) (either existing or self-collected)
  4. Problem(s) to work on
  5. Potential ways to solve problem(s)
  6. References

### Final Report

The project report describes and summarizes efforts on your selected topic. It should be **no longer than ten pages** and should include at least the following information:

- Project title and group members
- Problem(s) you are working on
- Dataset(s) you are using

- Algorithms / techniques / approaches you proposed or tried
- What part of the project was implemented on your own
- Experimental results and analysis
- Future work
- Roles of your group members (if there are 3 or 4 people in your group)
- References
- Code of your project implementation (as a separate archive) or link to your project repository. If your code is not organized in a notebook with step-by-step instructions, please also provide a simple README document about how we (and others) can run your code.

## Potential Topics

You may make use of these potential topics as a starting point.

1. **Energy consumption optimization:** Develop a TinyML model to optimize energy consumption in IoT devices by adjusting settings or controlling power-hungry components.
2. **Object detection and classification:** Implement a TinyML model for detecting and classifying objects using data from cameras or other imaging sensors.
3. **Environmental monitoring:** Design a TinyML model that monitors environmental conditions, such as air quality or noise levels, and provides alerts or insights based on the data.
4. **Smart agriculture:** Develop a TinyML model for precision agriculture applications, including soil moisture monitoring, plant health assessment, or pest detection.
5. **Health monitoring:** Create a TinyML model to monitor and analyze vital signs or other health-related data collected from wearable devices.
6. **Emotion recognition:** Implement a TinyML model capable of recognizing emotions from facial expressions or speech patterns.
7. **Activity recognition:** Design a TinyML model that identifies various human activities (e.g., walking, running, sitting) using data from wearable sensors.
8. **Fall detection:** Develop a TinyML model capable of detecting falls in real-time using accelerometer and gyroscope data, which can be useful for elderly care or worker safety.
9. **Sleep tracking and analysis:** Develop a TinyML model to monitor and analyze sleep patterns using data from wearable devices or bed sensors.
10. **Indoor air quality monitoring:** Create a TinyML model to monitor and provide insights on indoor air quality using data from air quality sensors.
11. **Music genre classification:** Design a TinyML model capable of classifying music genres based on audio features.
12. **Smart irrigation control:** Design a TinyML model that optimizes irrigation scheduling for plants based on factors such as soil moisture, weather data, and plant

growth stages.

13. **Vehicle license plate recognition:** Create a TinyML model that reads and recognizes vehicle license plates using data from cameras, which could be useful for parking management or security applications.
14. **Robust TinyML models against adversarial attacks:** Develop a TinyML model that is resistant to adversarial attacks, ensuring the model maintains its accuracy and performance even when presented with adversarially crafted input data.
15. **Detecting adversarial examples in TinyML:** Design a TinyML model or algorithm capable of detecting adversarial examples in real-time. The goal is to identify inputs that have been maliciously crafted to deceive the model and trigger incorrect predictions, enabling countermeasures to be taken before the adversarial example affects the system's performance.