

Capstone Oral Exam

Requirements

- 20-24 slides
- 40-45 minute time slot
- 10% of final grade
- *What do we need to include in this??*

Marking Scheme

Problem Definition - 20

Design Choices, Analysis, and Decision - 50

Clarity and Concise Presentation - 15

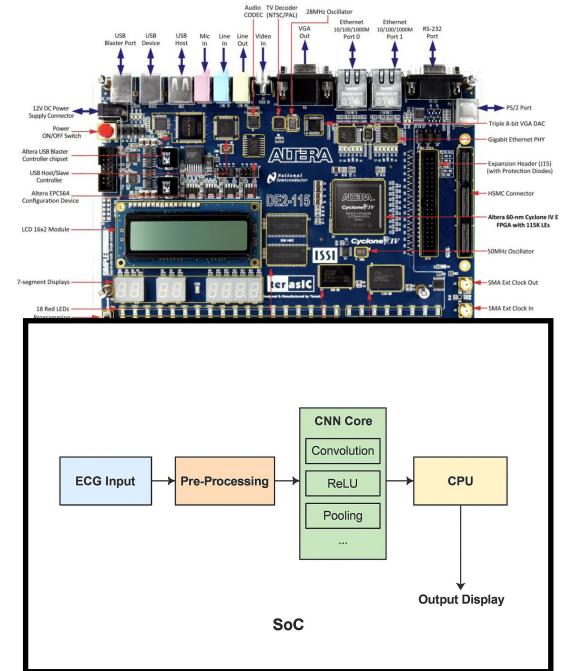
Preparedness for COE 70B (Implementation Plan) - 15

Individually Marked

Malcolm's 6 Slides (15 minutes)

Project Plan

- Divided work into logical, loosely coupled subteams, that can work concurrently.
 - CNN Model → Ayoub; SoC Team → Malcolm, Pierre; Simulation Component → Marly
- CNN Team
 - Deliverables: CNN-based architecture + model weights
- SoC Team
 - Deliverables: All SoC hardware that implements CNN model
- Simulation Component Team
 - Deliverables: simulation device that visually demonstrates our system
 - Implemented using FPGA to facilitate learning.



Neural Networks

- Last time I tried to learn about DL was in 2015. I kicked the can down the road after realizing the math that went into it.
- Worked through 6.S191 to understand basic of DL.
- Made some practice models: image classifier, digit classifier

The screenshot shows a Jupyter Notebook interface. At the top, there's a thumbnail for a YouTube video titled "Machine Learning Specialization by Dee...". Below the video thumbnail, a diagram of a "Fully Connected Neural Network" is displayed, showing two input nodes (x_1 , x_2) connected to a single hidden node, which in turn connects to a single output node. A list of characteristics for a fully connected layer includes: "Connect neuron in hidden layer to all neurons in input layer", "No spatial information!", and "And many many parameters!".

The main content area of the notebook shows the following text and code output:

```
[9]    ✓ 13.7s
...
... Malcolm's Digit Classifier Accuracy:
Accuracy for class: 0      is 99.4 %
Accuracy for class: 1      is 99.4 %
Accuracy for class: 2      is 98.6 %
Accuracy for class: 3      is 96.8 %
Accuracy for class: 4      is 99.6 %
Accuracy for class: 5      is 99.6 %
Accuracy for class: 6      is 99.6 %
Accuracy for class: 7      is 99.6 %
Accuracy for class: 8      is 99.6 %
Accuracy for class: 9      is 99.6 %
...
... CIFAR10 Image Classifier Accuracy:
Accuracy for class: plane is 52.5 %
Accuracy for class: car   is 61.4 %
Accuracy for class: bird  is 32.6 %
Accuracy for class: cat   is 37.8 %
Accuracy for class: deer  is 46.7 %
Accuracy for class: dog   is 63.2 %
Accuracy for class: frog  is 59.7 %
Accuracy for class: horse is 69.5 %
Accuracy for class: ship  is 57.2 %
Accuracy for class: truck is 67.2 %
```

On the right side of the notebook, there's a sidebar with the "MIT Deep Learning" logo and a thumbnail of a video titled "IntroToDeepLearning.com".

Neural Networks (Model)

- Read through a variety of papers to find a simple CNN-based architecture. Decided on this one: [ECG Heartbeat Classification: A Deep Transferable Representation](#)
 - 13 weight layers total: 1 initial conv + (5 blocks × 2 convs) + 2 FC = 13
 - Easy to implement in SW for DL noob. Hopefully implementable on hardware!
- [Current model](#) hosted here on Colab (might be some bugs!)
 - Attained 88% abnormal recognition, 92% normal recognition. Possibilities to improve here.

ECG Heartbeat Classification: A Deep Transferable Representation

Mohammad Kachuee, Shayan Fazeli, Majid Sarrafzadeh

Electrocardiogram (ECG) can be reliably used towards accurate categorization of heartbeat: a set of conditions on a dataset annotated for propose a method based on deep convolutional

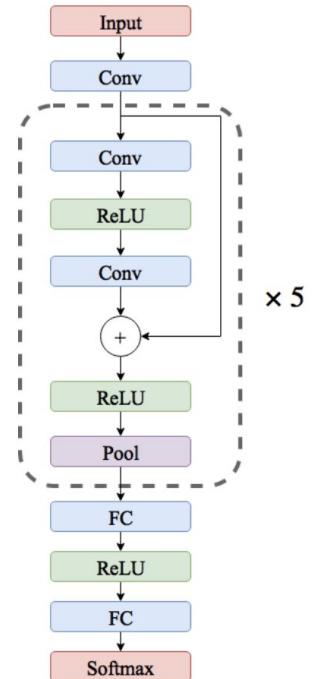
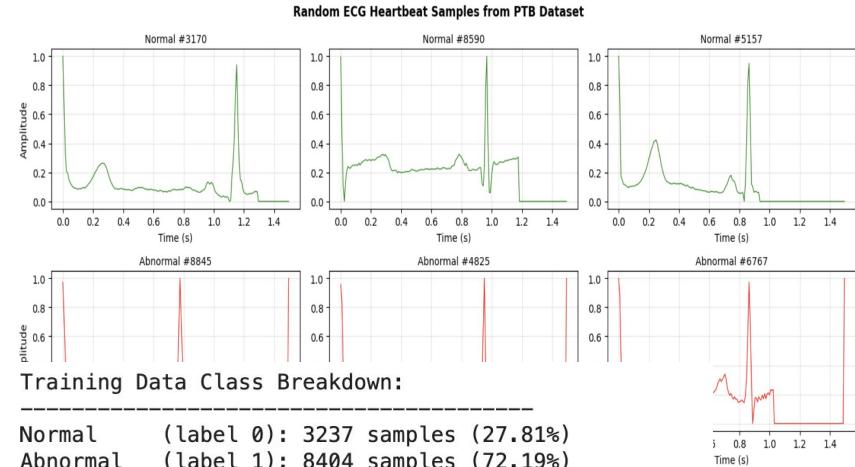


Fig. 2: Architecture of the proposed network.

Neural Networks (Data)

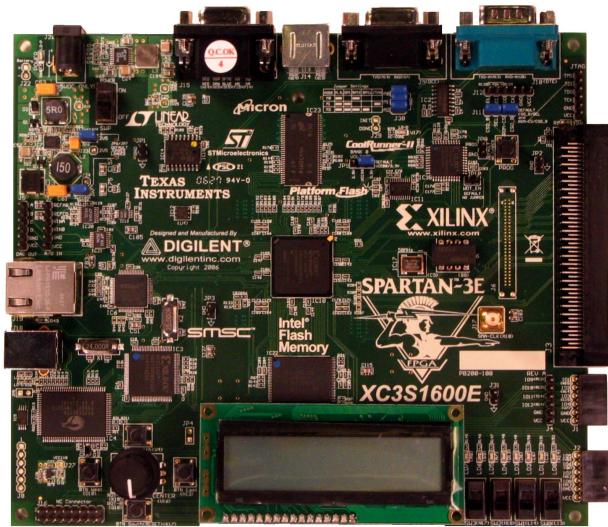
- Based on the Kaggle [heartbeat](#) dataset. Pre-processed data from MIT + PTB ECG datasets.
 - PTB: 125 samples/s, 188 samples each, normal/abnormal categories, single channel, 1 heart beat per csv row. Contains ~500 heartbeats.



Test Data Class	Diagnostic class	Number of subjects
Normal (label 0)	Myocardial infarction	148
Abnormal (label 1)	Cardiomyopathy/Heart failure	18
	Bundle branch block	15
Total: 2911 samples	Dysrhythmia	14
	Myocardial hypertrophy	7
	Valvular heart disease	6
	Myocarditis	4
	Miscellaneous	4
	Healthy controls	52

Hardware Implementation

- Xilinx Spartan 3E + ISE Design Suite
- Design Flow:
 - Quantize model, synthesize, route + place, iterate until things work smoothly.



Key Metric: % speedup in inference on FPGA vs. M1 Neural Engine

Next Steps

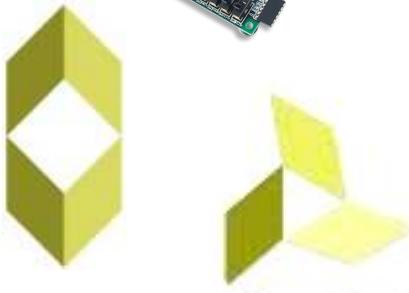
- Implement simple NN on FPGA using OSS design tools (e.g. hls4ml)
- Implement more advanced NN on FPGA platform
 - Iterate on CNN model + model quantization + HLS until this works very well.

Team is committed to completing significant work over the 3 weeks of
Winter Break

Pierre's 6 Slides

Hardware Decision

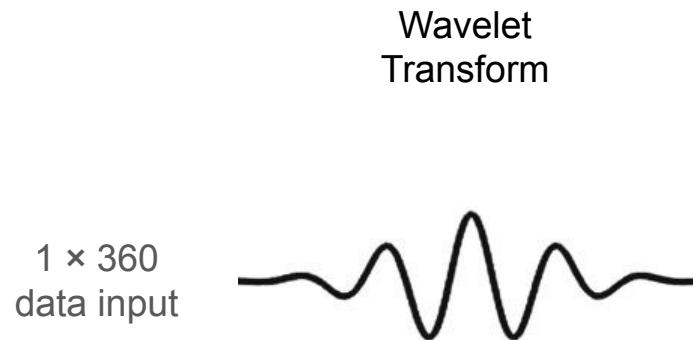
Spartan 3E XA3S500E



- 10,000 logic cells
- MicroBlaze softcore CPU
 - onboard flash memory holds up to 360 kB of C code
- Vivado Design Suite can synthesize a SystemC model into C and VHDL
 - Additional practice in SystemC gained from COE838
- I/O is _____

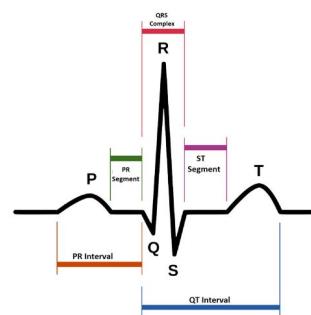
Dataflow

based on “ECG-based machine-learning algorithms for heartbeat classification”, [Nature](#)



Fractional Fourier Transform

$$F^{\alpha}(x(t)) = X_{\phi}(u) = \int_{-\infty}^{\infty} x(t)K_{\phi}(t, u)dt$$



CNN

- Normal
- Ventricular

DFT in VHDL

Ayoub's 6 Slides

ECG Arrhythmia Detection & My Project Contribution

- ECG arrhythmias occur when the heart beats irregularly (abnormal rhythm).
- Project goal: classify heartbeats as **Normal (N)** vs **Ventricular (V/E)**.
- Ventricular beats are clinically important because they appear wide and abnormal.
- My responsibility: **Develop Python-based CNN and dataset pipeline** for baseline classification.

Dataset Construction (MIT-BIH)

- Used MIT-BIH Arrhythmia Database
- Extracted around-R-peak segments (360 samples)
- 2 classes: Normal (N), Ventricular (V/E)
- Preprocessing: normalization, train–val split
- Output: `X_train.npy`, `y_train.npy`, `X_val.npy`, `y_val.npy`

CNN Design Options Researched

- Traditional ML (SVM / KNN / Random Forest)
- 2D CNN (treating ECG as an “image-like” input)
- 1D CNN (used)

Marly's 6 Slides

