CSCI 581 – Final ML Model Proposal Summary (fill in ALL answers below)

Project Title:	**Hardware-Aware Training Time and Throughput Prediction for CNNs on A100**
Model Type:	**ANN (Yes) RNN___ CNN(Yes)DQN___ Cluster___ Other: performance regressor**
Input:	**Text___ Image (Yes) Numeric (Yes)Sequence___ Vector___ Other_____**
Output:	**Detector___ Clasifier___ (#___) Predictor (Yes) Generator___ Game Ctrl___ Other_____**
Learning:	**Supervised (Yes) Unsupervised___ Reinforcement___ Semi-supervised___**

	Question	Answer	Notes	Points	Score
1a	What starter code for bottom-up or Tool will you use for Top-down for your ML model?	I will use a top-down approach with Python, TensorFlow/Keras, and the built-in CIFAR-10 dataset. My starter code will be a standard Keras CNN example for CIFAR-10 that I adapt to run on the cscigpu A100 and measure time per epoch.		10	
1b	Have you attached the starter code that compiles and/or runs OR an example of Tool use?	Yes, I will attach a minimal TensorFlow/Keras CNN script that trains on CIFAR-10 for a few epochs on GPU and prints the model summary plus average time-per-epoch. This will be the starter code I extend for my experiments.		10	
1c	How will you verify the ML model? PR, ROC, test set, training/validation, cross-validation, etc.	**two** models: • the CNN that learns CIFAR-10 • the ANN regressor that predicts training time I will verify the base CNN with a standard training/validation/test split on CIFAR-10 and report accuracy and loss curves. I will verify the performance regressor		10	

		using a train/validation/test split over the collected configurations and use regression metrics such as MAE, RMSE, and R^2, plus a predicted-vs-actual time-per-epoch scatter plot.			
1d	Have you included an example of training data you plan to use (e.g., on Google drive, from Roboflow, from Kaggle, from TensorFlow, etc.)?	For the base CNN, I will use the CIFAR-10 image dataset provided by TensorFlow/Keras (50,000 training images, 10 classes). For the performance regressor, I will generate my own CSV dataset where each row is one training job on A100, with columns such as depth, base filters, batch size, number of parameters, and measured time-per-epoch and images/second.		10	
2	What method(s) do you plan to use to speed-up training to make it run in parallel? – A100? Google Colab? Other?	I plan to run all main experiments on the cscigpu A100 using Python and TensorFlow/Keras, taking advantage of GPU data-parallel training with larger batch sizes. I may do small debugging runs on my personal machine or Colab, but the main speed-ups and measurements will come from running on the A100 GPU.		10	
3	What mathematics (numerical) method is involved?	The base CNN uses convolution operations, ReLU activations, softmax output, and cross-entropy loss optimized with gradient descent (Adam). The		10	

		performance regressor is a dense ANN trained with a regression loss (MSE/MAE) to approximate a nonlinear mapping from model/hyperparameter features to time-per-epoch. I may also compare against a simple linear regression baseline.			
4	What machine will you test on? Please provide output showing # of cores, GPU co-processing if any, and memory.	I will primarily test and train on the department's cscigpu A100 server (NVIDIA A100 GPU with many CPU cores and large system memory) and will include lscpu, nvidia-smi, and free -h output to document cores, GPU, and RAM. I may also run small tests on my personal laptop just for debugging.		10	
5	How do you plan to deploy and/or test your model?	I will "deploy" the models as a reproducible Jupyter notebook and Python scripts that can be run on the training machine. Testing will use train/validation/test splits for both the CNN and the performance regressor, with metrics (accuracy for CIFAR-10, MAE/RMSE/R^2 for the regressor) and visualizations (training curves and predicted-vs-actual plots). I will also include a simple function or notebook cell where a user can enter a CNN configuration and see the predicted time-per-epoch and throughput.		10	

6	**Why is this of interest to you?**	I am very interested in ML infrastructure and hardware-aware training (e.g., GPUs and custom accelerators like AWS Trainium/Inferentia), not just model accuracy. This project lets me connect what we learned about ANNs and CNNs in CSCI 581 with real GPU training behavior by building an empirical cost model for CNN training on an A100. It directly supports my goal of working on ML systems where understanding training time, throughput, and resource usage is as important as building accurate models.	10	
7	**What do you see as your biggest challenge to complete this?**	The biggest challenge will be running enough CNN configurations on the A100 to collect a high-quality dataset while managing GPU time and logging everything consistently. It will also be challenging to choose the right features so that the performance regressor generalizes well to unseen configurations and to organize the whole pipeline (experiment grid, data collection, regression training, and analysis) within the final project timeline.	10	
Proposal Score (must be 70 or above to be accepted)			**100**	
Are you willing to present? (circle one)		**YES**	**NO**	
Presentation Day: Mon 12/8 _____, Wed 12/10 _____, Fri 12/12 Yes , Mon 12/15 ____, Overflow Final Time 12-15-12-18_____ (choice: 1st and 2nd)				
Note: Presenting gets you out of making a video and if you present, your lowest quiz score will be replaced with a 100! In addition to make-up quiz. Follow Link to record choices.				

Please consider the following when you write a simple proposal that answers the questions above and ***be as concise as possible*** with your answers above. You can elaborate in Notes.

For the last 2 questions, please provide a few sentences describing why this is a significant demonstration of what you have learned in CSCI 581 and why it is challenging for you.

1) What ML Tools or code do you plan to start with or develop?
   a) Keras, TensorFlow, YOLOv8 Ultralytics, MATLAB, OpenCV?
   b) If you plan to build something bottom-up, do you have starter C++ code?
   c) If you plan a creative/research project, do you have an LLM, GAN, etc. to start with as an example?

   Ans) I plan to use **Python with TensorFlow/Keras** in a top-down way. I will start from a standard Keras **CNN example for CIFAR-10** as my base model and adapt it to run on the cscigpu A100 and log time-per-epoch and images/second. On top of that, I will develop a

small **dense ANN regressor** that takes model/hyperparameter features as input and predicts training time and throughput.

2) What method to you plan to use to make training scalable?
    a) C++ or Python for A100 – cscigpu
    b) Python with Google CoLab
    c) C++ or Python with Personal equipment

Ans) I will use **Python on the A100 cscigpu server** to take advantage of GPU acceleration and larger batch sizes for faster training. I may do small debugging runs on my personal machine or Colab, but all systematic experiments and measurements for the project will be done on **A100** so I can study realistic training performance.

3) What type of machine learning model will you use?
    - Regression (linear or polynomial)
    - K-means or DBscan
    - SVM
    - Dense input ANN
    - CNN
    - RNN
    - Etc.

Ans) I will use both **CNNs** and a **dense input ANN** in a supervised setting. The CNNs will learn the CIFAR-10 image classification task, and then I will train a dense ANN **regression model** that predicts training time-per-epoch and images/second from numeric descriptors of the CNN architecture and hyperparameters (depth, filters, batch size, number of parameters, etc.).

4) What machine will you test on and have you already tried training or testing an example or starter model?
    - E.g., my own hardware and here's the CPU cores (# with lscpu), memory (free -h), and other key resources required
    - A100 – cscigpu

Ans) I will primarily test on the **cscigpu A100** machine. I will include lscpu, nvidia-smi, and free -h output in my report to document CPU cores, GPU, and RAM. I will run a small starter CNN on CIFAR-10 on cscigpu early in the project to confirm that TensorFlow/Keras is working correctly on the A100 and that I can measure time-per-epoch and images/second. I may also run tiny tests on my personal laptop just for quick debugging.

5) How do you plan to deploy and test your ML model?
    - PR
    - ROC

- Cross validation
- Training, validation strategy and test set
- Deployment to embedded or web? Or just test on training machine?

Ans) I will deploy the project as a **reproducible Jupyter notebook and Python scripts** that run on the training machine. The CIFAR-10 CNNs will be tested using a standard **training/validation/test split** with accuracy and loss curves. The performance regressor will be tested with a train/validation/test split over the collected configurations and evaluated using **regression metrics** (MAE, RMSE, R^2) plus a **predicted-vs-actual time-per-epoch plot**. I will also include a simple notebook function where a user can enter a CNN configuration and see the predicted time-per-epoch and throughput. This is a deployment on the training machine, not a web app.

6) Why is this of interest to you?
    - Re-work of past problems you want to master? – if so, what would you do differently in comparison to the original problem?
    - How will this help you achieve learning objectives?

Ans) This project is interesting to me because it connects **deep learning models** (CNNs on CIFAR-10) with the **hardware behavior** of training on a real GPU (A100). In CSCI 581 we learned how to build and train ANNs and CNNs, but here I go a step further and use those models to generate data about training time and throughput, then train a second ANN to predict that performance. It is a significant demonstration of what I have learned because it uses course topics (ANNs, CNNs, supervised learning, training/validation/test, evaluation metrics) in a more **systems-oriented, hardware-aware** way that is directly relevant to real ML infrastructure work.

7) What do you see as your biggest challenge?
    - Finding and defining training, validation, and test sets for data used
    - Coding your Python or C++ code for model creation, compilation, and training
    - Finding test data and assessing performance in terms of PR, ROC, etc.
    - Advanced training methods like hyperparameter grid search and cross-validation
    - All of the above will make it a challenge – hyperparameters, model configuration, training, validation, testing

Ans) The biggest challenge will be **collecting enough high-quality data** on the A100 while managing time and resources. I need to design a grid of CNN architectures and hyperparameters, run many training jobs, and log time-per-epoch and throughput consistently, which combines hyperparameter tuning, experiment management, and GPU usage. A second challenge is choosing the right **features and regression model** so the performance predictor generalizes well to unseen configurations and is accurate enough to be useful. Bringing all of this together—CNN training, GPU profiling, regression modeling, and clear analysis—will be a challenging but valuable way to apply what I learned about ANNs, CNNs, training, validation, and evaluation in CSCI 581.

Goals:

MIN:

Train multiple CNN configurations on CIFAR-10 on A100, log time-per-epoch and images/second, and analyze how depth, filters, and batch size affect training performance.

TGT:

Train a dense ANN regression model that predicts time-per-epoch (and possibly images/second) from configuration features (depth, base filters, batch size, number of parameters), with reasonable R^2 and a clear predicted-vs-actual plot.

OPT:

Add accuracy–time tradeoff analysis, what-if examples for choosing configurations under a time budget, and discuss how the same method could be extended to other accelerators (e.g., AWS Trainium/Inferentia) for job-time estimation.