



Development Standards & Practices Used

List all standard circuit, hardware, software practices used in this project. List all the engineering standards that apply to this project that were considered.

Development Standards:

* ISO/IEC 12207
* TCP/IP
* FTP Circuit Standards
* High and low level schematics
* Parts lists
* simulation results
* OpenCL
* XML
* Linux Standard Base
* Unified Modeling language
* Message Passing Interface

Summary of Requirements

List all requirements as bullet points in brief.

* Use live-feed camera input
* Extract Frame Using Open-CV libraries
* Use Tensorflow & Keras pre-trained model on large dataset
* 6 x faster processing than existing algorithm
* 99% above accuracy on open-closed eye detection
* Reduce the model to a acceptable rate

Applicable Courses from Iowa State University Curriculum

1. CPRE 281
2. CPRE 288
3. COMS 352
4. EE 224
5. EE 324

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

1. Machine Learning
2. Acceleration on Embedded Systems
3. Camera interface
4. OpenCV library

Table of Contents

[1 Introduction 4](#_heading=h.1fob9te)

[1.1](#_heading=h.3znysh7) Acknowledgement 4

[1.2](#_heading=h.2et92p0) Problem and Project Statement 4

[1.3](#_heading=h.tyjcwt) Operational Environment 4

[1.4](#_heading=h.3dy6vkm) Requirements 4

[1.5](#_heading=h.1t3h5sf) Intended Users and Uses 4

[1.6](#_heading=h.4d34og8) Assumptions and Limitations 5

[1.7](#_heading=h.2s8eyo1) Expected End Product and Deliverables 5

[2. Specifications and Analysis 6](#_heading=h.3rdcrjn)

[2.1](#_heading=h.26in1rg) Proposed Approach 6

[2.2](#_heading=h.lnxbz9) Design Analysis 6

[2.3](#_heading=h.35nkun2) Development Process 6

[2.4](#_heading=h.1ksv4uv) Conceptual Sketch 6

[3. Statement of Work 7](#_heading=h.44sinio)

[3.1 Previous Work And Literature 7](#_heading=h.2jxsxqh)

[3.2 Technology Considerations 7](#_heading=h.3j2qqm3)

[3.3 Task Decomposition 7](#_heading=h.1y810tw)

[3.4 Possible Risks And Risk Management 7](#_heading=h.4i7ojhp)

[3.5 Project Proposed Milestones and Evaluation Criteria 7](#_heading=h.2xcytpi)

[3.6 Project Tracking Procedures 7](#_heading=h.1ci93xb)

[3.7 Expected Results and Validation 7](#_heading=h.3whwml4)

[4. Project Timeline, Estimated Resources, and Challenges 8](#_heading=h.qsh70q)

[4.1 Project Timeline 8](#_heading=h.3as4poj)

[4.2 Feasibility Assessment 8](#_heading=h.1pxezwc)

[4.3 Personnel Effort Requirements 8](#_heading=h.49x2ik5)

[4.4 Other Resource Requirements 9](#_heading=h.2p2csry)

[4.5 Financial Requirements 9](#_heading=h.147n2zr)

[5. Testing and Implementation 9](#_heading=h.3o7alnk)

[5.1](#_heading=h.23ckvvd) Interface Specifications 9

[5.2](#_heading=h.ihv636) Hardware and software 9

[5.3](#_heading=h.32hioqz) Functional Testing 9

[5.4](#_heading=h.1hmsyys) Non-Functional Testing 10

[5.5](#_heading=h.41mghml) Process 10

[5.6](#_heading=h.2grqrue) Results 10

[6. Closing Material 10](#_heading=h.vx1227)

[6.1 Conclusion 10](#_heading=h.3fwokq0)

[6.2 References 10](#_heading=h.1v1yuxt)

[6.3 Appendices 11](#_heading=h.4f1mdlm)

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

# 1 Introduction

## Acknowledgement

Special Thanks of gratitude to JR Spidell from Collins Aerospace who gave us the golden opportunity to do this wonderful project.

## Problem and Project Statement

Problem:

Hypoxia, Fatigue, Strain Doubles would cause pilots to crush their airplane, huge losses economically and not to mention they put their life in danger.

Project Statement:

Machine Learning algorithm has been developed to detect the blink rate on pilots, based on research, several blink patterns are related to Hypoxia, Fatigue and strain Doubles. Machine learning is loaded onto a fast processing edge device(FPGA) to detect such fatigue patterns. Our goal is to improve the existing machine learning algorithm, in terms of accuracy and efficiency by accelerating the algorithm from both software and hardware perspective.

## Operational Environment

The end product will go into an aircraft cockpit environment. There are a lot of environmental constraints associated with that.

However, we are not designing the mechanical or electrical systems here. So this specific project doesn/t have environmental constraints.

Instead, the constraints we have to work with are related to latency, accuracy, and memory consumption.

## Requirements

Functional Requirements:

Improve accuracy of the algorithm -- not less than 97%

Reduce the latency of the algorithm -- reduce latency by 75%

Reduce the memory requirements -- reduce memory requirement by 75%

Economic/Market requirement:

1) material costs, 2) development costs

Material costs can be 1) development materials 2) the material cost to make the product.

From a development material perspective, we have to live within the constraint of $1K for materials that have already been purchased.

The material cost to make the project is not a constraint for this project, since we are focused on the algorithms.

From a development cost perspective, the primary driver is the cost of people's time. Our budget for this is the allowable time we have between now and your final report.

Environment Requirements:

There are no specific environmental requirements for this project.

## Intended Users and Uses

Our intended users for the end product will be the pilot and help them protect their life from unexpected health problems which may make the warcraft out of control. However, we are the middle part of the whole large project so the constraints we have to work with are related to latency, accuracy, and memory consumption.

## Assumptions and Limitations

|  |  |
| --- | --- |
| Assumptions | For the end product, the accuracy of the algorithm will not be less than 97%; the latency of the algorithm will reduce latency by 75%; memory requirement for reducing memory by 75%. |
| Limitations | We don’t have users for our project at this time. The limitation is the optimization of accuracy, latency and the memory should get the client requirements. |

## Expected End Product and Deliverables

Expected End product and Deliverables:

* Optimized Machine Learning Algorithm

We will improve the existing machine learning algorithm using xilinx IP (optimization framework), tensor board, camera interface, to find the optimized efficiency while not losing significant accuracy. The technique we use such as finding optimized hyperparameters, pruning, parallel computing in programmable logic, in hope to achieve a 6 times faster detection capability.

# 2. Specifications and Analysis

## Proposed Approach

Functional Requirements:

Improve accuracy of the algorithm -- not less than 97%

Reduce the latency of the algorithm -- reduce latency by 75%

Reduce the memory requirements -- reduce memory requirement by 75%

Non-functional Requirements:

Economic/Market requirement:

1) material costs, 2) development costs

Material costs can be 1) development materials 2) the material cost to make the product.

From a development material perspective, we have to live within the constraint of $1K for materials that have already been purchased.

The material cost to make the project is not a constraint for this project, since we are focused on the algorithms.

From a development cost perspective, the primary driver is the cost of people's time. Our budget for this is the allowable time we have between now and your final report.

Environment Requirements:

There are no specific environmental requirements for this project.

Standards:

1. Circuits standards:

○Schematics

○Parts lists

○Simulation results

1. Digital Design standards:

○High and low level schematics

○Simulation results

1. Software standards:

○Source code

○In-line comments

○External documentation

## Design Analysis

discuss what you did so far:

---We setted up the anaconda environment, Xilinx tool suite, Ultra96-v2 board, experiment in Tensor board, and made our first progress on making RGB color space pictures into grayscale, which gives us significant improvement in time efficiency.

Did it work? Why or why not?

--- It worked, because for pictures that are 3 color space requires a lot more computation, when we can reduce it to grayscale without losing accuracy, grayscale gives 42% faster performance,

What are your observations, thoughts, and ideas to modify or continue?

---Machine learning algorithm acceleration can be done from both software and hardware, the real significant acceleration only comes from writing VHDL and programming at register level, however software acceleration is just as important. The best acceleration comes from a deep understanding of the algorithm from a software perspective, then designing a custom chip, ASIC for that specific machine learning algorithm. .

Results:

---Improved existing machine learning algorithm, by reducing color space from RGB to grayscale, we see an improvement in time efficiency by 42%

Strength:

---Huge improvement on existing machine learning algorithm in terms of speed

weakness:

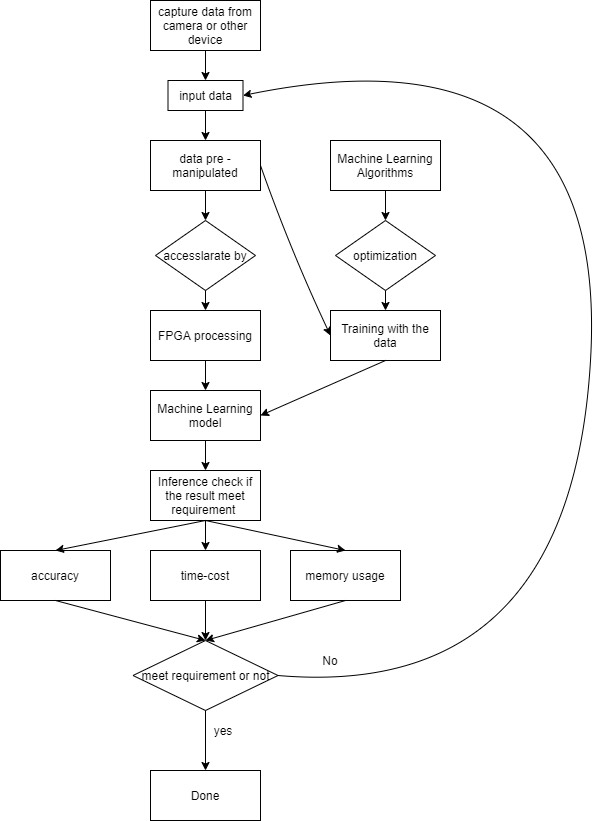
---A lot more acceleration is left undone, pruning, optimized hyperparameters, hardware acceleration etc.

## Development Process

Agile is the process we choose to follow, the rationale being agile lets us fast iterate and it is highly adaptable to changes. We have a baseline to test efficiency and accuracy, and we experiment with different techniques, if a technique seems promising, we integrate it.

## Conceptual Sketch

*Use a system-level conceptual sketch diagram to describe the concept for your approach/design. Describe the modules in your design (dependency/concurrency of modules, interfaces, architectural overview, etc.), and module constraints tied to the requirements.*



# 

# 3. Statement of Work

## 3.1 Previous Work And Literature

Include relevant background/literature review for the project

–  If similar products exist in the market, describe what has already been done

–  If you are following previous work, cite that and discuss the **advantages/shortcomings**

–  Note that while you are not expected to “compete” with other existing products / research groups, you should be able to differentiate your project from what is available

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

BackGround:

-- Although floating points are a good choice for handling the small updates that during Convolutional Neural Network training, the resulting parameters can contain too much redundant information.

-- BNN, in which some of the arithmetic involved in computing outputs are represented in only a single bit. Kim and Smaragdis[1] has published their work on the full binarization with a preset portion of the synapse having zero weight, and all other synapses with a weight of one. They report 98.7 % accuracy with a fully-connected network on MNIST dataset.

-- DoReFa-Net by Zhou et al. [2] explores reduced precision utilizing both the forward pass and the backward pass, he observes that this opens opportunities for training the neural networks on FPGAS. Their results include the best-case ImageNet Top-1 accuracies of 43% for full and 53% for partial binarization.

-- Major advantages of using BNN is the drastic improvement in efficiency. However, we can also expect a large amount of drop in accuracy. For our project, we can utilize the inference on the FPGA with either full or partial binarization.

## 3.2 Technology Considerations

Highlight the strengths, weakness, and trade‐offs made in technology available.

Discuss possible solutions and design alternatives

--- Major strengths using quantization, pruning and hyper-parameter tuning is the ability to inference at a faster speed, also taking less memories at the same time.

--- However, this could somehow affect our inferencing accuracies with unseen data. We are also facing the classic accuracy-computation tradeoffs.

--- One alternative is using k-fold validation to validate with our pruning, quantization, image-pre manipulation and hyper-parameter tuning. So the test set can give unbiased feedback.

## 3.3 Task Decomposition

In order to solve the problem at hand, it helps to decompose it into multiple tasks and to understand interdependence among tasks.

-- Image pre manipulation

-- Hyper-parameter tuning

-- pruning

-- Quantization of weights

-- FPGA design for acceleration

The tasks are taking different angles in effort to reduce the operations needed for the inference. On the training side we pre manipulate images, fine tune the layers, pruning the unnecessary connections; On the infereing side, we quantize the neural network and map onto the FPGA.

## 3.4 Possible Risks And Risk Management

Include any concerns or details that may slow or hinder your plan as it is now. These may include anything to do with costs, materials, equipment, knowledge of area, accuracy issues, etc.

-- Coronovirus delaying our camera daughter card

-- Reducing computation may influence our inference accuracies

## 3.5 Project Proposed Milestones and Evaluation Criteria

What are some key milestones in your proposed project? Consider developing task-wise milestones. What tests will your group perform to confirm it works?

-- Key milestones: Delopying algorithm on FPGA;

Obtain baseline algorithm metrics(memory,speed,accuracy)

Improve algorithm to surpass baseline metrics

-- Tests: Run the modified algorithm on ULTRA96-v2(FPGA) with the same measurement of time, memory, accuracy. Comparing the newly obtained results with baseline metrics.

## 3.6 Project Tracking Procedures

What will your group use to track progress throughout the course of this and next semester?

-- GNATT chart.

-- Google Doc

## 3.7 Expected Results and Validation

What is the desired outcome?

-- Improve speed by 50 %

-- Keep the accuracy above 97 % on unseen data

How will you confirm that your solutions work at a **High level**?

-- Comparing the newly obtained metrics with baseline metrics; on the same dataset

# 4. Project Timeline, Estimated Resources, and Challenges

## 4.1 Project Timeline

• A realistic, well-planned schedule is an essential component of every well-planned project

• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

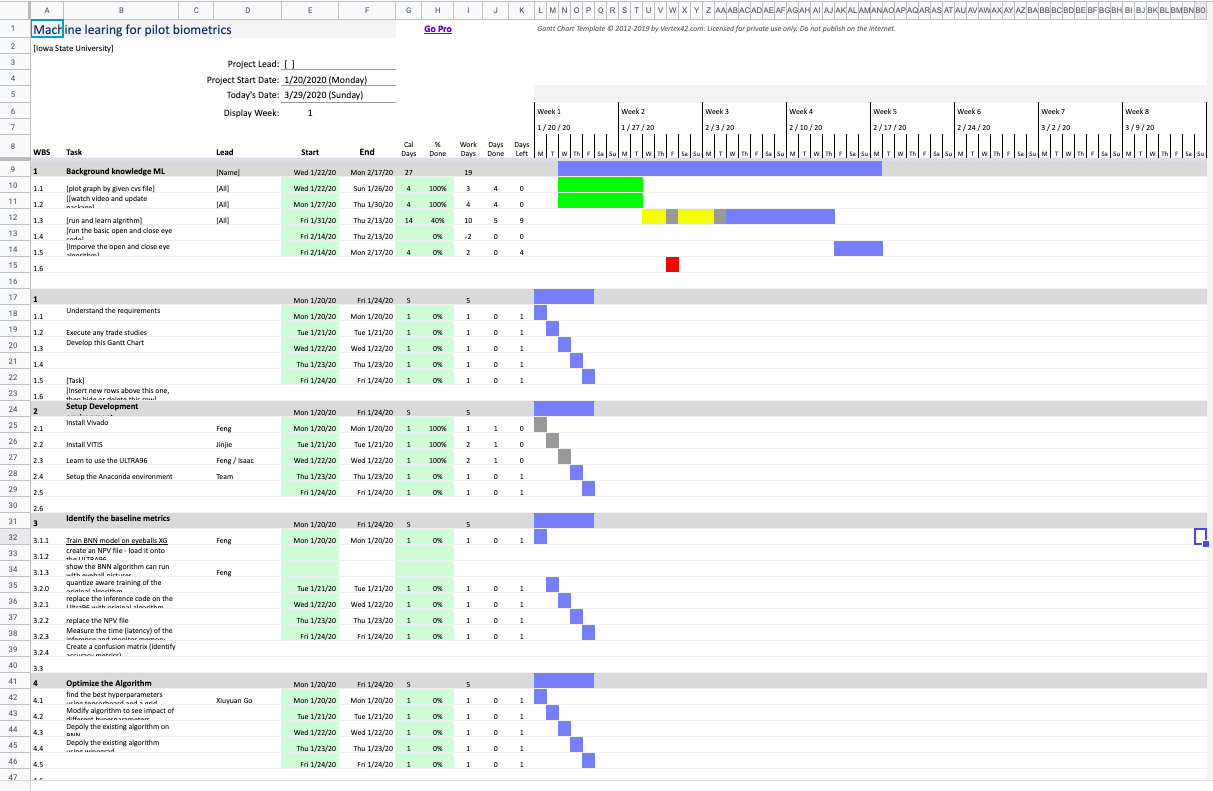
– Start with a Gantt chart showing the tasks (that you developed in 3.3) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Completely compatible with an Agile development cycle if that’s your thing

How would you plan for the project to be completed in two semesters? Represent with appropriate charts and tables or other means.

Make sure to include at least a couple paragraphs discussing the timeline and why it is being proposed. Include details that distinguish between design details for present project version and later stages of project.



-- paragraphs: The timeline is being proposed to be agile and adaptable to changes as we get familiar with the technology and may want to experiment/follow different methods to achieve our goal and capture results based on a well-defined guideline, and obtain our result(metrics) in a reliable manner.

## 4.2 Feasibility Assessment

Realistic projection of what the project will be. State foreseen challenges of the project.

-- We will make some significant improvement on latency, memory usage while keeping a pretty good accuracy.

## 4.3 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be based on the projected effort required to perform the task correctly and not just “X” hours per week for the number of weeks that the task is active

|  |  |  |
| --- | --- | --- |
| Task | Explanation | Requirements |
| Pruning | Prune the network | 143 hrs |
| Hyperparameter Tuning | Fine tune the network | 144 hrs |
| Sparse Filters | Using sparse filters | 147 hrs |
| FPGA design | FPGA hardware design | 143 hrs |
| Camera Trade Study | Study trade off on camera options | 142 hrs |
| BNN implementation of algorithm | Binarize the weights for algorithm | 143 hrs |
| Camera FPGA integration | Integrate Camera with FPGA | 144 |

## 4.4 Other Resource Requirements

Identify the other resources aside from financial, such as parts and materials that are required to conduct the project.

-- Camera Daughter

-- Ultra96-v2

-- Xilinx License

## 4.5 Financial Requirements

If relevant, include the total financial resources required to conduct the project.

-- Financial support for hardware purchase

# 5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)  
2. Define the individual items to be tested  
3. Define, design, and develop the actual test cases  
4. Determine the anticipated test results for each test case 5. Perform the actual tests  
6. Evaluate the actual test results  
7. Make the necessary changes to the product being tested 8. Perform any necessary retesting  
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you’ve determined.

Needed test for our project includes:

Unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements

1. Individual items to be tested for our project includes:
2. Ultra96-v2 functionality test
3. Software-hardware integration test
4. unit testing for pruning, quantization

Some actual test cases include unit testing for each of our software components, for instance we have thoroughly tested each of our components, ie the part for metrics measurements, quantization such before we merge everything together for integrity testing.

Challenges:

Unstable environment

## Interface Specifications

– Discuss any hardware/software interfacing that you are working on for testing your project

Hardware interfacing: ultra96-v2 board, MIPI Camera, camera daughter card

Software interfacing: IP network protocols, software drivers for the peripheral devices

## Hardware and software

–  Indicate any hardware and/or software used in the testing phase

–  Provide brief, simple introductions for each to explain the usefulness of each

-- software drivers for ultra96-v2, this is useful because we want to make sure drivers is not causing us connections issues

-- IP network protocols for communication between host and embedded development board, this is useful because we want to eliminate possibility of bugs that may introduced by protocols communication issue

## Functional Testing

Examples include unit, integration, system, acceptance testing

--- Unit test for software that measures metrics

--- Integration test for running the metrics on ultra96 ARM processor

--- Acceptance testing for running new algorithm and make sure the metrics is better than the baseline

## Non-Functional Testing

Testing for performance, security, usability, compatibility

--- Performance testing for keep the memory usage low

--- Compatibility test for running frameworks that support ARM53 processor architecture

## Process

–  Explain how each method indicated in Section 2 was tested

–  Flow diagram of the process if applicable (should be for most projects)

-- We tested the software drivers by connected the ultra96 to different host computer and make sure the opperatate driver was supported in different operating system

-- we tested different connectivity protocols utilizing different connection protocols like uart, usb etc.

## Results

– List and explain any and all results obtained so far during the testing phase

* –  Include failures and successes
* –  Explain what you learned and how you are planning to change it as you progress with your project
* –  If you are including figures, please include captions and cite it in the text

• This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place

-**Modeling and Simulation**: This could be logic analysis, waveform outputs, block testing. 3D model renders, modeling graphs.

-List the **implementation Issues and Challenges**.

--- All connect and drivers test indicates successes

--- What I learned is that we should stuck with the official supported operating system and software so the likelihood of running to unknown bugs are low

--- One implementation issues is that some of the existing frameworks supports older version of Xilinx tools that has been archived

# 6. Closing Material

## 6.1 Conclusion

Summarize the work you have done so far. Briefly reiterate your goals. Then, reiterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

--- We have set up the metrics, improved the algorithm by more than 80% in terms of latency, some more improvement can be done in terms of accuracy.

--- Explore hardware acceleration and utilizing the FPGA advantage to harvest more computation power and improve our algorithm further

## 6.2 References

This will likely be different than in the project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).   
[1] M. Kim and P. Smaragdis. Bitwise neural networks. [26] CoRR, abs/1601.06071, 2016.

[2] S. Zhou, Z. Ni, X. Zhou, H. Wen, Y. Wu, and Y. Zou. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. CoRR, abs/1606.06160, 2016.

## 6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.