UNIVERSITY OF SOUTH CAROLINA

PROJECT PROPOSAL

UNIVERSITY OF SOUTH CAROLINA

COMPUTER SCIENCE AND ENGINEERING

CSCE-585: Machine Learning Systems
Project Proposal: Visual M.L. System for Identifying and Minimizing Civilian Fatality in Urban War Zones

---

Authors:

Logan Nall (github.com/Clnall),
Walter Pach (github.com/waltster),
Vera Svensson (github.com/Lux-Vera)

September 6, 2022

Contents

1 Project Repository

The repository for our team can be found at https://github.com/waltster/CSCE-585. This repository is used for collaborating, issue tracking, and updating/version control.

This proposal can be identified by the GitHub tag: **ProjectProposal**.

2 Problem & Interest

With the proliferation of automated systems deployment in firms and organizations around the world, there is a demand for automated systems that produce both efficient and cost-effective solutions to problems previously requiring human intervention. In recent years, the application of artificially intelligent systems has provided the solution for many organizations in the form of models and systems that are able to intelligently predict and respond to data. This introduces many concerns and opens the possibility for deployment in novel situations and sectors. The application of A.I. systems is not limited by industry but rather will impact markets uniformly as the technology continues to evolve. Firms are presented with the problem of keeping pace with changing market conditions, availability of relevant A.I. systems, and quality of A.I. models. As firms and organizations begin to adopt A.I. technology, competition that spurs further adoption will occur. This causes early concerns to be paramount in sectors that directly influence human life.

With the advancements made in object detection and its deployment at the edge, the military applications of A.I. systems have increased. The subject of our concern is the use of A.I. systems in automated or supervised targeting systems involved in the use or training of weapons systems. This includes both unmanned and manned, both unsupervised and supervised weapons systems. In order to preserve human life in this context while maintaining the cost-effectiveness and efficiency of these A.I.-enabled systems, we hope to outline both the necessity and the feasibility of introducing new control mechanisms in this environment.

In this project, our goal will be to develop an A.I. model that is enabled to differentiate between common civilian and military vehicles, including classifications such as land vehicles, airplanes, helicopters, and drones. The problem presented includes both the issue of accuracy when human analysts apply characterization to targeting, as well as the ability to audit the success of the system with a clearly defined confidence metric.

Given the conditions of the Russia-Ukraine war and the escalating international tensions, paired with the rapid deployment of developmental weapons, we find it of particular interest to develop a built-in capability to identify and avoid civilian casualty.

3 Background Literature / Related Work

There are documented cases where A.I. systems have been deployed alongside or to replace human operators, including in recent conflict [1]. This is of particular interest for highlighting the increasing role that M.L. systems are playing in production-level application.

CNN stands for Convolutional Neural Network and in "A Fast Military Object Recognition using Extreme Learning Approach on CNN" researchers look at combining ELM standing for Extreme Learning Machine to make up for the pitfalls of CNN [2]. The goal of this research is to maintain the accuracy of CNN but to alleviate its long training times, have a smaller resource need, and having comparable speed as needed for field use. Previous works with CNN and ELM have been done in the past achieving an accuracy of close to 95% albeit with long processing time and great commitment of resources while being trained. Data collected during the training time showed that CNN and ELM saved roughly 20% on training time however utilizing more CPU and RAM. The outcome of needing more resources was predictable as compared to just running CNN; the researchers found that it would take an additional 79% more resources, particularly for the CPU, for running models from both ELM and CNN. What researchers would come to conclude from using the two is the benefit of maintaining accuracy. In the test with CNN and CNN together with ELM, the pair had a 1% higher accuracy while just CNN had 3% greater results in the normal model test data. Through the five testing iteration CNN and ELM would achieve .86 accuracy while CNN .89 showing that 3% difference.

Contrasting CNN and ELM individually, CNN would continue to produce better results in more complicated identifications such as helicopters over the single layer using ELM. But the need to combine the two stems from CNN's back-propagation leaving more to be desired in time saving capabilities.[2]

TABLE VI. TABLE COMBINATION OF CNN AND ELM ACCURACY, PRECISION, AND RECALL RESULTS

Accuracy	**0.87**	
Kelas	**Precision**	**Recall**
Military Helicopter	0.78	0.77
Armored Car	0.77	0.81
Military Tank	0.86	0.76
Military Jet	0.75	0.80
Military Ship	0.93	0.90
Pistol	0.89	0.92
Military Rifle	0.95	0.90
Grenade	0.87	0.89
Military Box	0.86	0.79
Military Knife	0.89	0.88
Military Helmet	0.98	0.98
Military Binoculars	0.89	0.94
Military Boot	1.00	0.95
Military Bag	0.97	0.97
Army	0.99	0.95
Non-Military	0.64	0.81
Avg Micro	**0.88**	**0.88**
Avg Macro	**0.88**	**0.88**

Figure 1: CNN and ELM's Results [2]

Information is vital to formulating decisions and the military has to make many decisions from the repetitive to the new. Machines can handle large data spectacularly well. In the paper "Human-AI Cooperation to Benefit Military Decision Making" authored by Karel van den Bosch and Adelbert Bronkhorst, they speak to the problem of bound rationality as Herbert Simon would refer to it where people are "limited in their cognitive processing abilities when they have to deal with complex problems" despite however rational they may be. Machine learning is already implemented and

being improved to be able to suggest solutions. Allowing less time wasted sorting through data and more time utilized on developing the best solution.

From technology developing, drones are the latest to be introduced to the battlefield. Often used for surveillance, they have classification mainly that of weigh, payload, and range.

Class 1 and 2 are capable of having weapons because of their size and capable range. Class 3, 4, and 5 for closer range surveillance. Being able to identify civilian vehicles is not only important for drones but also for satellites. These methods of getting visual confirmation are the most frequently used.

A methodology for detecting military objects instead of civilian objects is the proposition for a multi-level CapsNet architecture [3]. The idea of this method is to have an image go through the following:

1. Convolutional Layer
2. Primary Capsule Layer
3. Class Capsule Layer
4. Softmax Layer

Then, having an output as a result of crossing through all the layers. The Convolutional Layer can be multiple layers as the input image is being scanned for features by the application of kernels. An interesting idea to keep account of is if the image is not black and white as in such an instance then the kernels used need to be 3D arrays to factor in depth juxtaposition to black and white's 2D kernels. The output from the Convolutional Layer goes into the Primary Capsule Layer that will formulate an array of feature maps to be reshape into a number of vectors by a dimension. An important step here is managing the vectors to be between 1 and 0 since probability will be used to determine object's presence, this step being called the "squash function". Class Capsule Layer is quite complicated where features for identification and classification will become the output. This is done through the process of route by agreement which seems to be a division of the output of the Primary Capsule Layer being contrasted against each other in three groups for a smaller output. This output is what the Softmax Layer ultimately weighs against each possible military object class providing the result. [3]

4 Data

The M.L. system will be developed iteratively along the same methodology as the source. We plan on training our model on several different data sets to be able to evaluate how well it performs. Our model will begin being trained on a simple data set of different car makes and models. We plan to use the CompCars data set which is a data set from 2015 containing labeled images of cars.[4] This data set is slightly lacking, since we want out model to be able to identify more than just cars. However,

we think that it would be of interest to able to evaluate how well the model can differentiate between similar vehicles. Additionally, this will allow our boilerplate code to be developed iteratively and identify the libraries that we need to incorporate.

After our system has been adapted and trained on the sample car data, we will use a collection of military and civilian vehicles. We plan on using the "Military and Civilian Vehicles Classification" data set, available as FOSS [5]. It contains 6772 images of both civilian and military vehicles of different sorts. This will provide both the labelling and the images to train our final model.

As we continue the development of our model we might discover faults with these data sets, or find other data sets of interest. We are open to add, or switch, data sets along the way if we find it fitting.

5 Methodology & Integrations

5.1 Technological Methodology

Our initial evaluation of the technology available suggests that our M.L. system will require many components to achieve full functionality. For the modelling and data representation, the best solution for our problem statement appears to be an object-detection model trained specifically to detect the presence of vehicles. This model will then be re-trained on another labelled data set with the nuance of military/civilian differences.

Our initial idea of the system is using static images that are fed through the object-detection model and outputs a classification with a confidence metric. We will achieve this through training a TensorFlow model. Our second iteration will be to use a video feed that is split into its static frames when movement occurs, with the same modelling process taking place.

5.2 Developmental Methodology

Our team's development methodology begins with a concern for the ease of delivery and adaptation of the A.I. model(s) and supporting infrastructure over time. This means that the process for both building, collaborating, and testing code must be able to scale appropriate to the needs, and that the system must function similarly as the scale of the deployment increases. We incorporate an iterative methodology, where each iteration produces an increase in accuracy, efficiency, or reliability.

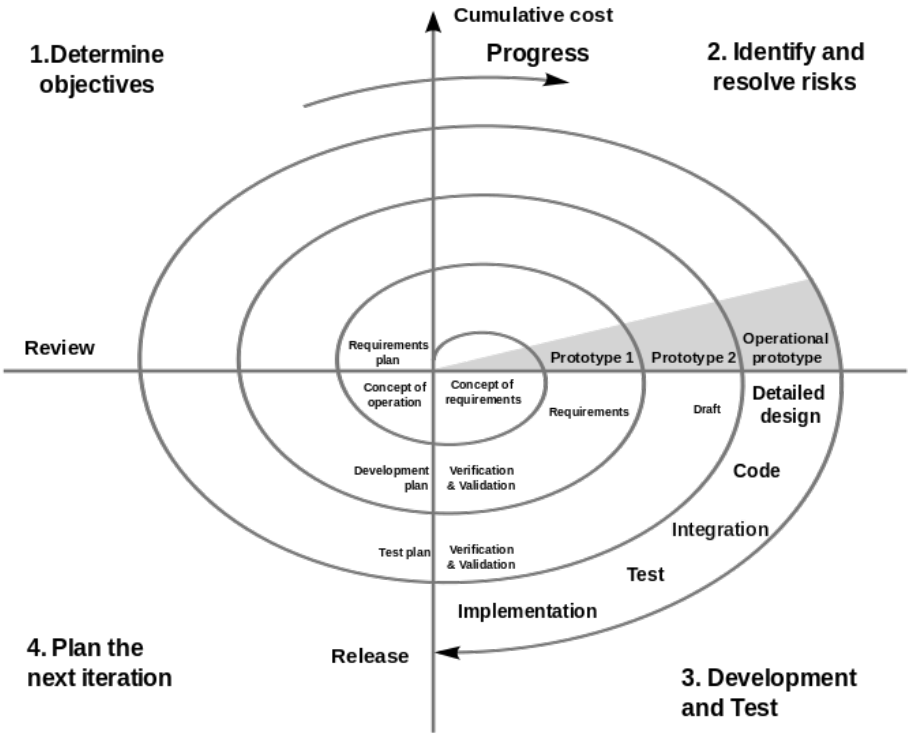

Figure 1: Spiral Development

For source management, our team will use GitHub which offers support for issue tracking, collaboration, and version management. Additionally, we will use GitHub's branching ability to manage the deployment and deployment of different stages of the source. Organization of the code will be along `scopes`, meaning that the source's intention and ability will expand to a new, defined `scope`. Contributions will be organized on a per-contributor branching scale.

Our team will use Trello for administrative task management such as deadlines, team updates, and communication about the state of the project. Trello provides a Kanban view of "cards" which will allow our team to organize tasks in different states.

5.3 Experimental Methodology

Our team will use a six-stage experimental methodology. This includes the initial development of the boilerplate to wrap the system, including the Jupyter Notebook and the various utility functions for downloading content and models. The methodology focuses on the same incremental attitude towards progress.

1. Begin with writing a boilerplate to train models.
2. Train a model for identifying the difference in car makes.
3. Write a test framework to evaluate and provide stats on efficiency.
4. Train a model for the vehicle identification.
5. Run the experiment and gather results.
6. Evaluate results.

6 Evaluation Methods

6.1 Development Evaluation

To make sure our code is structured and unified we plan on using GitHub Actions workflow.[6] GitHub Actions is a tool that help us create an automated process that will build and test our code. This will make testing of the code easier and quicker since we reduce the manual test burden. We plan on creating a workflow that is triggered when a developer pushes to the repository. The purpose of this workflow will be to lint the code. Since we code in Python we will use *Pylint* and *Flake8*.[7]

It is possible to have several workflows and they are highly configurable, but we will be starting of with only one. We prefer to start of small and create one effective, yet simple, workflow that will aid us in the development process. Spending to much time on workflows will simply not be worth it, since this project has a rather short time frame.

6.2 AI System Evaluation

6.2.1 Qualitative Evaluation

With the M.L. system, we expect to receive results in the raw format of object identification and tagging on a static image. However, this data can be further analyzed *en masse* by applying statistical methods for how the model performed compared to the known state. For example, images with known military vehicles ranging from 1-10 can be fed through the system and the identification results analyzed in a spreadsheet and graphed. This will give both a visual identifier that the system was able/unable to identify the vehicles and a visual graph of the data collectively across tests.

6.2.2 Quantitative Evaluation

The M.L. system will provide quantitative results by providing statistical accuracy and probability measures for its ability to correctly identify a civilian or military vehicle. This will be conducted as a broad test with several tens or hundreds of test images. With the known value for civilian/military vehicles, standard equations can be applied using values such as:

- True Positive
- False Positive
- True Negative
- False Negative

By using these, the system can be evaluated for its performance and generate an accuracy value. This value can then be used as the range of the function of the M.L. system. The model and system can then be adjusted to improve this value, causing the function (M.L. system) to perform better.

7 Limitations

The M.L. system our team will develop has several clear limitations, both deterministic and indeterministic. The relevant variables that describe the limitations on the system are:

1. The extent and variability in vehicles that the model is trained on, including the variability posed by regime changes and organic development.
2. The implementation of the data source and its photographic clarity.
3. The processing capabilities of the device on which the analysis is being conducted (i.e., the processing power available may significantly influence the confidence metric)

The relevant limitations posed by this system are:

- Additional infrastructure must be constructed in order to use the confidence metric as a variable in cost-value analysis when targeting human beings.
- Analysis and insight is limited by the training data provided, with clear analytical leaps when the vehicles cannot be referenced to vehicles in the data set
- Does not limit or eliminate the ability for adversaries to use civilian vehicles as a shield, requiring human interactions
- Both civilian and military vehicles have different designs around the world. It's likely that the data sets we use are heavily weighted towards the American, or other large nations, design of vehicles. We deem the work load of creating our own data set, with a more even distribution of vehicles from around the world, to be to high. Therefor accuracy results are likely to differ depending on the what nation the vehicle belongs to.

References

[1] Kahn J. A.I. drones used in the Ukraine war raise fears of killer robots wreaking havoc across future battlefields. Fortune Magazine; 2022. [cited 5 September 2022]. Available at https://fortune.com/2022/03/29/artificial-intelligence-drones-autonomous-weapons-loitering-munitions-\slaughterbots-ukraine-war/. pages 4

[2] Hari Surrisyad W. A Fast Military Object Recognition using Extreme Learning Approach on CNN. IJACSA. 2020;11(12). [cited 5 September 2022]. Available from: https://thesai.org/Downloads/Volume11No12/Paper_27-A_Fast_Military_Object_Recognition.pdf. pages 4, 5

[3] Janakiramaiah B, Kalyani G, Karuna A, Prasad LVN, Krishna M. Military object detection in defense using multi-level capsule networks; 2021. [cited 5 September 2022]. Available from: https://link.springer.com/article/10.1007/s00500-021-05912-0#citeas. pages 6

[4] Yang L, Luo P, Loy CC, Tang X. A Large-Scale Car Dataset for Fine-Grained Categorization and Verification. CVPR; 2015. [cited 5 September 2022]. Available at http://mmlab.ie.cuhk.edu.hk/datasets/comp_cars/CompCars.pdf. pages 6

[5] Gupta P, Pareek B, Singal G, Rao DV. Military and Civilian Vehicles Classification. Mendeley Data. 2021. [cited 5 September 2022]. Available from: https://data.mendeley.com/datasets/njdjkbxdpn/1. pages 7

[6] Docs G. About workflows. GitHub Inc.; 2022. [cited 5 September 2022]. Available at https://docs.github.com/en/actions/using-workflows/about-workflows. pages 9

[7] Rawat AS. Enhancing Code Quality With Github Actions. Medium. 2020. [cited 5 September 2022]. Available from: https://medium.com/swlh/enhancing-code-quality-with-github-actions-67561c6f7063. pages 9