Friend or Foe: Multi-Modal Military Target Identification

Andrew Jeon • Bassam Halabiya • Naif Ganadily • Zachary Saunders
University of Washington
EE P 567: Machine Learning for Cybersecurity
Professor Radha Poovendran • Lead TA Qifan Lu

Roles and Responsibilities

We constructed our custom dataset together in equal parts, taking images mainly from Google, following the split below to create our initial 1100 image dataset.

- Andrew - US/RU Navy (250 Images) + Landscape/Urban "Null" (25 images)
- Bassam - US/RU Marines (250 Images) + Paintball "Null" (25 images)
- Naif - US/RU Army (250 Images) + Desert "Null" (25 images)
- Zachary - US/RU Airforce (250 Images) + Civilians "Null" (25 images)

Bassam took a leadership role in dataset construction, leveraging his expertise with Roboflow and dataset creation. He oversaw the quality control process for our dataset, meticulously providing feedback for refinement upon submission. He also managed most of the data augmentations on Roboflow to increase our dataset size to 2,652 images. With the dataset ready, we divided responsibilities for developing the Machine Learning model. Andrew, Bassam, and Naif collaborated on image classification into RU_military or US_military classes while Zachary focused on developing the Challenge Response System. Andrew, Bassam, and Naif experimented with YOLOv8 architectures to achieve and deliver initial detection/classification results. More specifically, Andrew employed YOLOv8Nano, Bassam utilized YOLOv8Small and YOLOv8Medium, and Naif experimented with YOLOv8Large and YOLOv8Extra Large. After initial modeling, Bassam and Naif led the refinement and optimization of the models to improve overall performance.

Regarding the written documentation, the project's inception was credited to Zachary, who took the lead position in drafting the proposal while incorporating feedback from the team. As for the checkpoint and final project reports, the team divided responsibilities equally among team members. For the checkpoint report, Andrew wrote about additional experiments, Bassam summarized preliminary conclusions, Naif outlined the milestones, and Zachary covered the main bottlenecks encountered. For the final report, Andrew focused on roles & responsibilities, references, and code documentation. Zachary drafted the problem statement and our proposal for tackling it. Naif contributed to the YOLOv8 architecture and Segment Anything Model (SAM). Finally, Bassam provided insights into experimental results & analysis as well as future work. To provide proof of concept, Bassam and Naif developed two applications to showcase real-world scenarios. Naif's development of a user-friendly web application using Streamlit showcased the accuracy and robustness of the YOLOv8x for recorded video and image analysis. On the other hand, Bassam integrated the challenge-response system with the AI model to showcase the suitability and reliability of the YOLOv8s model for real-time applications.

Context & Problem Statements

War is unpredictable; despite months of planning, steadfast discipline, and constant communication, any fighting force is only the sum of its parts, fallible and imperfect humans. Physical exhaustion, emotional distress, and unfamiliar environments notwithstanding, during combat soldiers are faced with a choice: to shoot a target or not to shoot. Superficially this proposition appears trivial: if the target is an enemy, fire, otherwise do not. However, it is not so. On the modern battlefield, combat occurs from hundreds of yards away, with rifles aimed at soldiers wearing camouflage, belonging to different factions, and organized in guerrilla formations, intentionally designed to challenge target identification. With varying degrees of trickery at play, mistakes do happen. Friendly fire and unnecessary casualties due to mistaken target identification have become a facet of warfare. “Many Americans [are] shocked to learn that 23 percent of all [American] casualties in the Gulf War were from [American] Weapons.” (Shrader) Military engagements can be won or lost by a factor far less than 23%, and therefore should there be a method by which soldiers may positively identify friend from foe, such technology would have a monumental impact on the success of their mission objectives.

With a clear need for a more effective target identification system, we explored existing solutions, analyzed standard issue equipment, and reviewed military doctrine, all in an effort to not only develop a solution that performed well but also seamlessly integrated with existing battlefield technologies. Through this research, we discovered that the most feasible solution for target identification involved visually capturing images through a camera, analyzing the captured images, and reporting to the user, the soldier in this case, the end result: is the target friend or foe? With this plan developed, more questions arise. Where do we place the camera in relation to the soldier? How does the camera detect friend and foe, and what factors will be used in this decision? How can such a system be protected and secured from adverse use? Beginning with the camera placement, many solutions were considered, however it was ultimately decided that two possible placements provided the optimal line-of-sight necessary to capture the target properly: the camera should either be attached to the soldier's helmet akin to a Microsoft HoloLens apparatus or attached to the optic of the firearm similar to a thermal imaging sensor. Although both solutions have their merits, having the camera affixed to the firearm allows the soldier to remain behind cover, being protected from enemy fire, while having the firearm and camera sensor engaged on the targets. With the camera in place, we then reviewed possible methods for target identification. The core problem with target identification is that of diversity; targets can be young, old, hairy, clean-shaven, tall, short, standing, crouching, etc. With so many variables, it is not feasible to develop a manual solution, instead, we must employ the use of AI. “AI is inherently a decision-making technology: it offers opportunities to automate many tasks relating to learning and devising solutions.”(Verganti et al., 2020) With an AI solution, our system would be capable of identifying its own significant features, which optimize the friend-or-foe determination, features that we have not manually defined. These features are developed in a “black-box” manner whereas we do not have visibility into exactly what features are in play, however, we can influence the AI model via the training data provided. As such, selecting proper training data is paramount. We chose to develop our dataset from images sourced online, augmented, and labeled according to the military faction that they contain.

With the AI model classifying friend and foe, we must consider adversarial efforts to defeat or co-opt the system. Should our system have any success, we can assume that the enemy forces will invest time and effort to neutralize its effectiveness to regain the upper hand. It then follows that we enhance the AI system with a secondary security factor. We again revisited our research to develop an enhancement solution that would be natural to our soldiers but also provide an added layer of defense. Echoing the challenge-response system already used in forward operating bases to verify the legitimacy of a detached soldier reentering a friendly installation, we developed a challenge-response system (CRS) that automatically confirms the suspected friendly as friendly and draws suspicion to a foreign soldier attempting to confuse the AI system. Whereas the AI system is attached to the firearm optic, the CRS has two parts: a directional antenna and a software-defined radio (SDR). Additionally, all soldiers' uniforms will now be equipped with embedded radio receivers and transponders. Much like an optic system, the antenna and SDR will point at a target in-plane with the firearm. When aimed at a target, the SDR will issue a challenge radio signal towards the direction of the target. If aimed at a friendly, the transponder will hear the signal and respond, confirming that they are indeed friendly. Although this second factor provides the confirmation we desire, it also introduces a larger attack service for the enemy to combat. The most obvious challenge with all this radio traffic (challenge, response, challenge, response, over and over across the radio waves) is the detectable signals being emitted. Should the enemy have radio towers or other equipment on the outskirts of their facilities, they would trivially be able to predict an attack and perhaps even gather the number of incoming soldiers, information that will jeopardize the lives of our friendly soldiers. Therefore, we have also implemented a Stealth Mode in the CRS system that will temporarily quiet all radio traffic during times of sensitive missions. Furthermore, we are aware of the risk that spoofing provides. If a friendly soldier is downed in combat, we need a means to eliminate that node from the CRS pool, this is handled via a timeout function. Furthermore, each CRS transponder uses a unique ID which makes spoofing even less possible.

With this multi-pronged approach to target identification, 3 overall problems needed to be addressed:

1. Collect, label, and ingest an image dataset of soldiers dressed in combat uniforms.
2. Design and develop an artificial intelligence (AI) algorithm that will correctly classify the images as either friendly or foe.
3. Design and develop an enhancement challenge-response system (CRS) that protects against adversarial efforts and can adapt to different mission objectives.

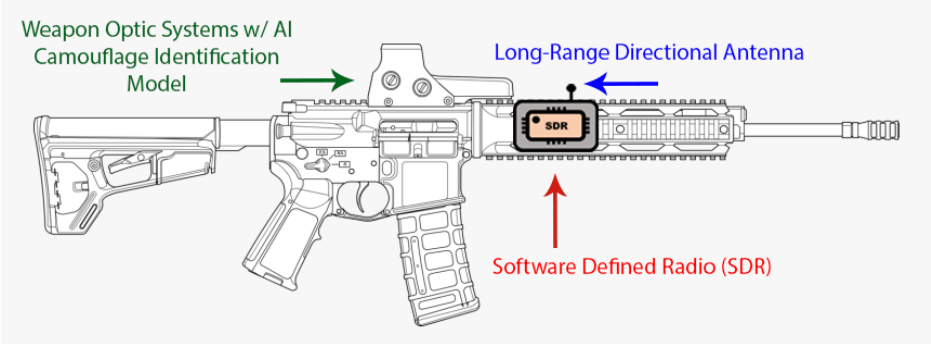

Figure 1: Firearm with AI Optic & CRS Components Installed
Image source provided in CRS Jupyter Notebook

SAM

The SAM (Segment Anything Model) presented by Meta offers a cutting-edge approach to image segmentation and annotation, At its core, SAM employs a unique network architecture that efficiently handles complex visual information. This involves a picture encoder transforming 1024x1024 images into comprehensive multi-channel feature maps, coupled with a prompt encoder that converts various prompts into vector embeddings. These components work together within a mask decoder layer to generate three distinct segmentation masks at varying levels of granularity, enhancing the model's precision even in ambiguous scenarios. Key to SAM's efficacy is its innovative use of loss functions, specifically the Intersection over Union (IOU) score, dice loss, and focal loss, which together optimize mask accuracy and address model performance on challenging examples. The architecture's efficiency stems from its ability to pre-compute and reuse image embeddings for different prompts, thanks to the independent processing within the mask decoder. Further distinguishing SAM is its incorporation of a vision transformer-based image encoder, facilitating the translation of images into embeddings with remarkable accuracy. This approach, inspired by foundational work in the field, enables the model to reconstruct images from a fraction of the original data, underscoring its robustness. The dataset fueling SAM spans manual, semi-automatic, and fully automatic stages, amassing a vast collection of annotated images that refine the model's capabilities through progressive training phases. SAM's architecture intricately blends image and mask encoders, along with innovative prompt encoding techniques for sparse and bounding box prompts, and a pre-trained CLIP model for text prompts. This comprehensive encoding framework ensures, a contextual understanding of visual information. The mask decoder's design employs multiple attention mechanisms to integrate prompt and image embeddings, assembling in high-quality segmentation masks. This system's effectiveness is evident across a range of tasks. In essence, SAM exemplifies a profound leap in image segmentation technology that we used for our annotations of the images through Roboflow's UI.

YOLOv8 for Friend or Foe

We utilized the YOLO (You Only Look Once) model, particularly version 8, YOLOv8 tackle the intricate task of differentiating between ally and enemy forces in military combat. This state-of-the-art algorithm is renowned for its effectiveness and precision in real-time object detection. It functions by dividing input images into a grid structure, where each grid cell is responsible for predicting bounding boxes and class probabilities. These predictions encompass bounding boxes situated at the center of each grid cell alongside confidence scores indicating object presence and class probability maps hinting at the object category within each cell.

YOLOv8 Architecture

The YOLOv8 model constructed on a Convolutional Neural Network (CNN) architecture converts input images into a sequence of output vectors. These vectors encapsulate class probabilities and bounding box parameters for carrying out object detection tasks. By abstracting the image through layers into a sophisticated representation this model generates output vectors for every grid cell based on high-level features derived from the input images. This transition from visual data to detailed classifications showcases the model's ability to rapidly and accurately interpret complex visual information.

Fine-tuning the YOLOv8

A crucial element of our project was in finetuning the YOLOv8 to cater to our requirements with a focus on distinguishing among different military personnel. We trained the model using our custom dataset to improve the accuracy of identifying friendly units and adversaries through bounding boxes and fine-tuning confidence scores to ensure our model could effectively differentiate between friends and foes in various operational scenarios.

Inference Process

During the inference stage, YOLOv8 utilizes maximum suppression to refine its predictions by reducing redundant bounding box detections and prioritizing highly confident predictions. This method is vital in settings where precision and quick decision-making can be critical for saving lives. We fine-tuned the model parameters incorporating localization and confidence losses to effectively penalize predictions and enhance overall performance and reliability.

Experimental Results and Analysis

In our computer vision-based machine learning project aimed at assisting soldiers on the battlefield, we conducted extensive experimentation with our custom dataset comprised of 1000 images representing the 4 branches of each military faction. Using our custom dataset allowed us to tailor it to the project's needs and requirements thereby enabling flexibility concerning annotation, segmentation, and class balance. This dataset was created, annotated, and deployed on Roboflow where it can be modified to accommodate various computer vision techniques, and manipulated to have user-specified training, validation, and testing split. This dataset is made more robust by adding augmentation and pre-processing techniques. For our project, we chose to implement random rotation, random Gaussian blur, salt & pepper noise, and horizontal flip to ensure the model can generalize better to unseen data. Our work focused on performing transfer learning with several pre-trained YOLOv8 models.

Before discussing the training and deployment of our model, it’s paramount to address the safety-critical aspect nature of the application we are developing. Given the context of military operations where instantaneous decisions can carry significant ramifications, the reliability, accuracy, and robustness of our system are essential to its success and the livelihood of those involved. False positives or misclassification of targets can result in compromising the mission or worse, friendly fire.

Dataset Details

The team compiled 1100 images of military soldiers sourced from the internet and each team member was responsible for collecting and annotating 275 images. The individual team member dataset comprised 125 images per specific branch for both military forces plus 25 null images. Each image was annotated using Roboflow’s annotation assistant, the Smart Polygon tool, to allow users the flexibility of training for segmentation, detection, or both. The team anticipated that adding augmentation techniques would enlarge the dataset to 4000+ images, however, this kind of extensive augmentation was reserved for the paid Roboflow memberships. The custom dataset contained 2652 images where 2328 images were assigned for training, 220 images for validation, and 104 images for testing. In Roboflow, several dataset

versions were created using the same source images. Table 1 outlines the three major dataset projects that the team generated and trained in Roboflow.

Table 1: Custom Datasets Published on Roboflow

Dataset Name	**Classes**	**Versions**	**Deployed Models**
Friend_or_Foe_Merged	RU_army, RU_airforce, RU_marines, RU_navy, US_army, US_airforce, US_marines, US_navy	1	1
Friend_or_Foe_Class_ Consolidation_ObjDet	RU_military and US_military	10	6
Friend_or_Foe_Class_ Consolidation_InstSeg	RU_military and US_military	2	2

Dataset issues

Our initial results for the machine learning model suffered from a few setbacks primarily due to issues with the dataset quality. The combination of these issues resulted in the poor performance of our computer vision model as it struggled to accurately distinguish friendly forces from foes. That said, due to time limitations and work priority, we were only able to address a few of the issues:

- Difficulty in acquiring images for the Russian military compared to the US military.
- Russian military images suffered from duplications, obscurity, and misalignments compared to the US military counterparts.
- Blurry images were exacerbated by the data augmentation techniques applied.
- Overly ambitious use of 8 classes to categorize annotations by military factions and branches
- Completeness optimization led to the inclusion of partial body parts for camouflage identification

Training Details

Several, but not all, detection and segmentation YOLOv8 models were utilized to accomplish this complex task of distinguishing between camouflage uniforms. The team took a meticulous approach to fine-tuning the YOLOv8 hyperparameters by taking into account epochs, learning rate, learning rate scheduler, loss function, optimizer, image size, and more. This effort was aimed at finding the balance between computational efficiency, accuracy, and resilience while assessing the applicability for real-time usage. The rigorous and iterative optimization process resulted in determining the ideal set of hyperparameters necessary for our system requirements.

Table 2: Ideal Hyperparameters for Training with YOLOv8

Parameter	**Value**	**Ultralytics Description** [2]
Epochs	25-50	Total number of training epochs. Each epoch represents a full pass over the dataset. Adjusting this value affects training duration & model performance.
Batch	16	Batch size for training, indicating how many images are processed before the model's internal parameters are updated.
Imgsz	640	Target image size for training. All images are resized to this dimension before being fed into the model. Affects model accuracy and computational complexity.
Optimizer	SGD	Options include SGD, Adam, AdamW, NAdam, RAdam, RMSProp etc., or auto for automatic selection based on model configuration.
Lr0	0.001	Initial learning rate (i.e. SGD=1E-2, Adam=1E-3). Adjusting this is crucial for the optimization process, influencing how rapidly model weights are updated.

Lrf	0.01	Final learning rate as a fraction of the initial rate = (lr0 * lrf), used in conjunction with schedulers to adjust the learning rate over time.
dropout	0.0 - 0.1	Dropout rate for regularization in classification tasks, preventing overfitting by randomly omitting units during training.

Evaluation Details

To evaluate our model's performance, we used recall, precision, and mAP. As a reminder, mAP refers to the mean Average Precision metric across all classes, while precision refers to how often the model predictions are correct (accurate positive predictions), and recall refers to the percentage of successfully identified labeled images (positive prediction completeness). The initial results demonstrate the learning process achieved in this project where we were able to eventually obtain a model capable of accurately distinguishing friend from foe with a mAP of 75.7%. These promising results pave the way for the possible incorporation into military applications via AI goggles or enhanced weapon optics/scopes. By experimenting with multiple YOLOv8 models, we were able to hone down on the hyperparameters that mattered the most. Table 3 below summarizes the test and experimentation results collected throughout this project.

Table 3: Transfer Learning Results with YOLOv8

Model	**Dataset**	**Hyperparameters**	**mAP50**	**mAP50-95**
YOLOv8n	Friend_or_foe_class_consolidation_objdet/10	Ideal (50 epoch)	0.711	0.536
YOLOv8s	Friend_or_foe_class_consolidation_objdet/7	Ideal (50 epoch)	0.736	0.581
YOLOv8m	Friend_or_foe_class_consolidation_objdet/8	Ideal (50 epoch)	0.773	0.636
YOLOv8l	Friend_or_foe_class_consolidation_objdet/5	Ideal (25 epoch)	0.371	0.307
YOLOv8x	Friend_or_foe_class_consolidation_objdet/9	Ideal (25 epoch)	0.757	0.613
YOLOv8x-seg	Friend_or_foe_class_consolidation_objdet/5	Ideal (25 epoch)	0.734	0.562

After completing our transfer learning experimentation with YOLOv8 which involved a significant amount of trial and error, we were able to streamline our approach by selecting two YOLOv8 models that performed well within our specifications. The first model, YOLOv8x, is much larger and has demonstrated greater accuracy when classifying uniforms making it the ideal choice for performing inference on recorded video and images. When it comes to image and video analysis, precision is preferred over speed, and this is where this model excels. However, due to its sluggish real-time performance and computational demands, this model is deemed unsuitable for real-time applications and therefore impractical for edge device deployment. On the other hand, the YOLOv8s model has proven to be highly efficient in real-time applications such as performing inference on live video stream frames. Although the smaller model is less accurate than YOLOv8x, its reduced computational overhead makes it ideal for deployments to edge devices. The YOLOv8s trained model is well suited to handle field operations usage where speed and accuracy are of equal importance. Using these two models, we can cater to various applications without having to sacrifice accuracy or speed.

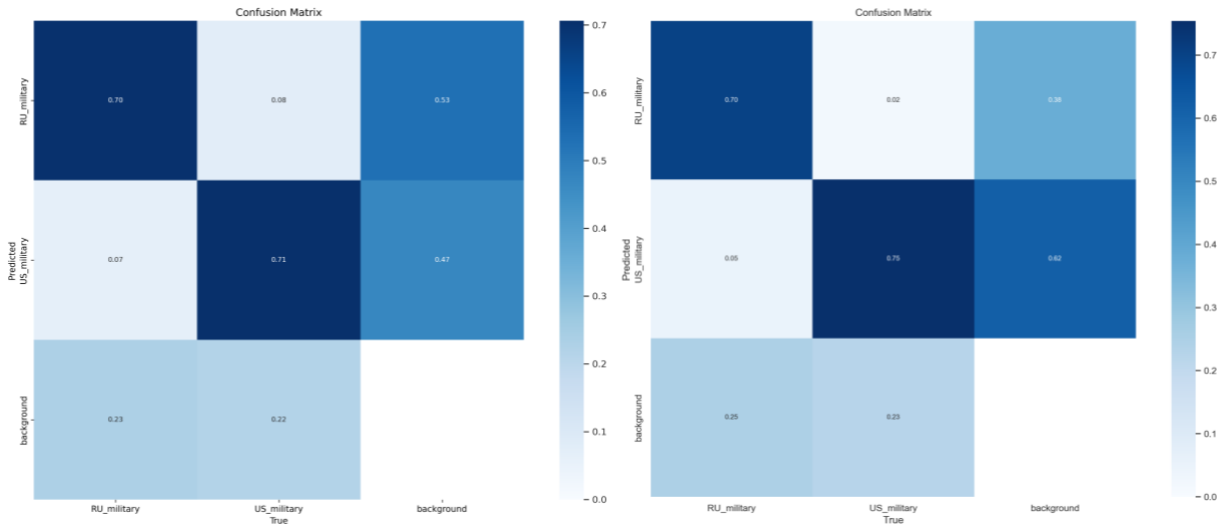

Figure 2: YOLOv8s (left) vs YOLOv8x (right) Confusion Matrices

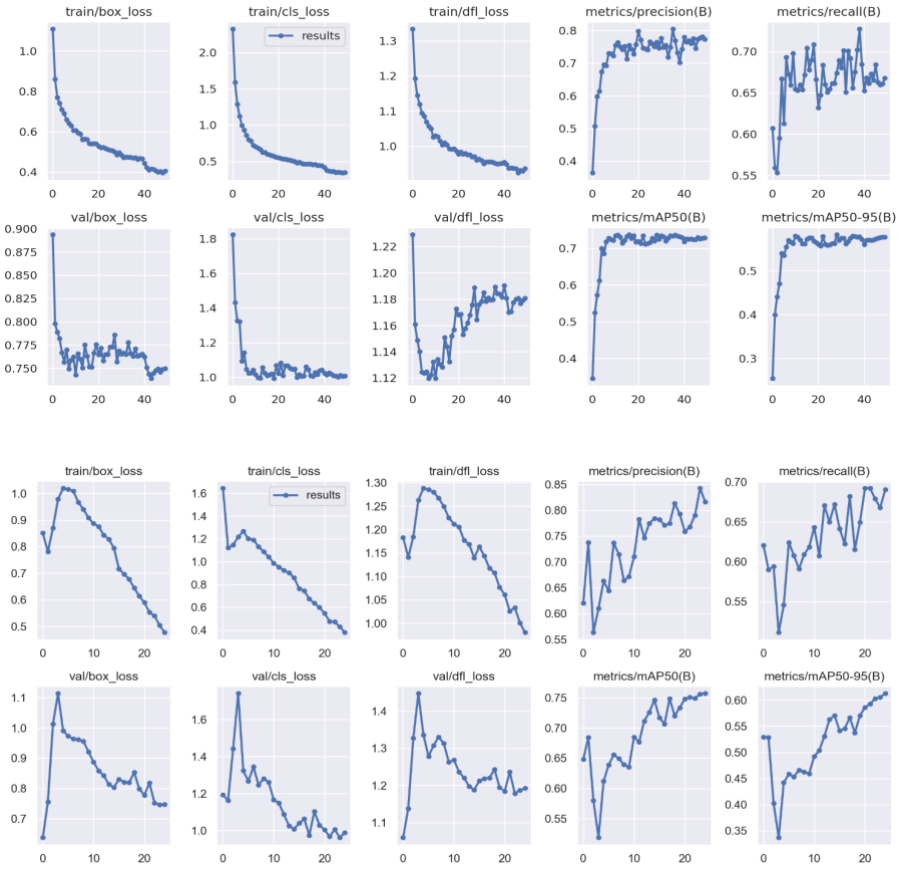

Figure 3: YOLOv8s (Top) vs YOLOv8x (bottom) Training Results

Figure 4: YOLOv8s (left) vs YOLOv8x (right) Prediction Validation Batch

Proof of concept

To showcase the capabilities of our system, two proof of concept products were developed. The first product is a user-friendly web application that leverages Streamlit and allows for real-time video processing, as well as image and recorded video analysis. This interface allows users to interact with the system upload live video feeds and quickly receive detection or segmentation results to differentiate between friends and foes. The application utilizes the YOLOv8 model for timely object detection and segmentation focusing on speed and accuracy. Although the current iteration of the web app lacks a challenge-response feature to authenticate identified targets due to Streamlit's limitations with external servers, this approach significantly enhances accessibility for end-users, particularly military personnel. This collaborative effort demonstrates how the fusion of advanced machine learning models with user-centric applications can enhance critical decision-making processes. The second and last proof of concept product aims to showcase the real-time capabilities of our system by integrating the CRS code with a Python script designed to perform inference on webcam video streams. This seamless integration with the webcam simulates realistic deployment scenarios, while leveraging socket connections for Challenge Response functionality, emulating software-defined radio setups. The script operates in AI mode, wherein the AI model processes each frame to detect and classify camouflage uniforms as friendly or hostile. The challenge-response system allows manual operation, with users issuing commands to challenge targets, prompting identity confirmation within a set timeframe. This demonstration not only highlights the accuracy and robustness of the model but also hints at potential integration with external software/hardware components.

Future Work

Our project will tackle key features and challenges for future growth and deployment. First, in order to capture a broader array of soldiers in the field and under varying lighting conditions, the team intends to

significantly enlarge the dataset. Additionally, we plan to take a meticulous approach when selecting the augmentation techniques for our dataset to not only enlarge the dataset further but also to allow the model to generalize better by addressing the issues noted in the experimental results and analysis section. Following this development, the computer vision model will be able to function more consistently in a range of real-world circumstances thanks to its improved resilience. We also aim to optimize the computer vision model by incorporating ensemble learning to draw additional insight from the data and increase classification accuracy and overall robustness. Looking beyond our current scope for military operations, we believe that our concept extends to law enforcement, disaster response, and perhaps search-and-rescue missions. Our project aspires to offer improved situational awareness by providing rapid identification and classification which in turn leads to safer and more effective operations in the future.

References

[1] "Yolov8: How to Train for Object Detection on a Custom Dataset." *YouTube*, Roboflow, 10 Jan. 2023, www.youtube.com/watch?v=wuZtUMEiKWY&t=962s.

[2] Burhan-Q, Laughing-q and glenn-jocher (2024) *Model Training with Ultralytics YOLO, Train - Ultralytics YOLOv8 Docs*. Available at: https://docs.ultralytics.com/modes/train/#train-settings (Accessed: 11 February 2024).

[3] Johnson, Reed. "Roboflow Inference: Effortless Local Computer Vision Deployment with Python." *Roboflow Blog*, Roboflow Blog, 8 Sept. 2023, blog.roboflow.com/inference-python/#roboflow-inference-local-deployment-made-easy.

[4] Shrader, C. R. (n.d.). The US Army War College Quarterly: Parameters. The US Army War College Quarterly: Parameters The US Army War College Quarterly: Parameters. https://press.armywarcollege.edu/cgi/viewcontent.cgi?article=1645&context=parameters

[5] SkalskiP. "YOLOv8 LIVE." *GitHub*, github.com/SkalskiP/yolov8-live/tree/master. Accessed 15 Feb. 2024.

[6] Ultralytics. "Ultralytics YOLOv8." *GitHub*, github.com/ultralytics/ultralytics. Accessed 17 Feb. 2024.

[7] Verganti, R., Vendraminelli, L. and Iansiti, M. (2020), Innovation and Design in the Age of Artificial Intelligence. J Prod Innov Manag, 37: 212-227. https://doi.org/10.1111/jpim.12523

NOTE: *CRS Sources are listed in Jupyter Notebook, image sources are listed in Github, with the sources below in reference only to this document.*

Appendix

Project Repository:
https://github.com/Naif-Ganadily/Friend-or-Foe-Multi-Modal-Military-Target-Identification