Detecting Patterns of Life Using Deep Learning

Javier E. Garza, Patrick H. Rupp
Lockheed Martin Corporation
Fort Worth, TX
Javier.Garza@lmco.com,
Patrick.H.Rupp@lmco.com

Anastacia MacAllister, PhD
General Atomics Aeronautics Systems
Poway, CA
Anastacia.Macallister@ga-asi.com

ABSTRACT

Commercial and government organizations are now collecting 100 terabytes or more of overhead imagery via satellites and drones on a daily basis. Specifically, within the Department of Defense (DoD), analysts spend enormous amounts of time sifting through these data to detect events of interest, categorize them, and report them through the appropriate channels. Analysts are well-trained in their ability to sift through data, however, the amount of analysts available to perform this work is limited. The volume of the data is increasing rapidly, and as this increase continues, it will be more difficult for analysts to find the bandwidth to support this activity. As a result of this limited bandwidth and lack of equivalently increasing number of personnel, the DoD community will have to prioritize which data to analyze. This will inevitably lead to unintentionally missing significant events. As this trend continues, how can the DoD improve this process and alleviate the workload? One solution is to employ machine learning; more specifically, deep learning to the imagery to perform change detection as well as object detection and localization such that further methods for drawing higher-level insights on Patterns of Life can be enabled. Employing these methods could ensure that all of the data gathered could be analyzed without requiring more analysts. The paper will describe the background associated with Patterns of Life analysis and discuss in detail the development of a modified U-Net architecture adapted to do change detection for military applications with overhead imagery. The described architecture will show the DoD community a viable approach for dealing with the problem of scalability that comes with collecting increasing amounts of data. Additionally, the paper will explain how the method was tested with publicly accessible satellite imagery datasets, and finally, describe conclusions about the work in terms of warfighter applicability.

ABOUT THE AUTHORS

Javier Garza is an AI/ML Engineer and Program Manager at Lockheed Martin Skunk Works®. His work focuses on architecture, development, integration, testing, and deployment of machine learning algorithms into military platforms. He has more than ten years of experience in software engineering, open architecture technologies, flight test, and technical leadership. Mr. Garza has worked on Defense Advanced Research Project Agency (DARPA) programs such as System of Systems Integration, Technology, and Experimentation (SoSITE) and Target Recognition and Adaption in Contested Environments (TRACE). Additionally, he has worked on several Internal Research and Development projects at Lockheed Martin and NASA Mission Control Center (MCC) software modernization projects. Mr. Garza has a bachelor’s degree in Computer Science and a master’s degree in Software Engineering from the University of Texas at El Paso.

Anastacia MacAllister, Ph.D., is an AI/ML Solutions Architect for General Atomics Aeronautical Systems. Her work focuses on prototyping novel machine learning algorithms, developing machine learning algorithms using sparse, heterogenous or imbalanced data sets, and exploratory data analytics. Throughout her career she has provided key contributions in a number of interdisciplinary areas such as prognostic health management, human performance augmentation, advanced sensing, and artificial intelligence aids for future warfare strategies. This work has received several awards and commendations from internal technical review bodies. Dr. MacAllister has published over two dozen peer reviewed technical papers, conference proceedings, and journal articles across an array of emerging technology concepts. Dr. MacAllister holds a bachelor's degree from Iowa State University (ISU) in Mechanical Engineering. She also holds a Master's and PhD from ISU in Mechanical Engineering and Human-Computer Interaction.

Patrick Rupp is a Data Engineer at Lockheed Martin's Skunk Works®. Mr. Rupp specializes in developing data pipelines particularly for use in analytics problems ranging in size from small prototypes to enterprise solutions. Mr. Rupp is currently developing data pipelines for various AI/ML programs and has previously supported Lockheed's work in Defense Advanced Research Projects Agency's (DARPA) Systems of Systems Integration Technology and Experimentation (SoSITE) program and Air Combat Evolution (ACE). Mr. Rupp received his Master's in Data Science from Johns Hopkins University.

Detecting Patterns of Life Using Deep Learning

Javier E. Garza, Patrick H. Rupp
Lockheed Martin Corporation
Fort Worth, TX
Javier.Garza@lmco.com, Patrick.H.Rupp@lmco.com

Anastacia MacAllister, PhD
General Atomics Aeronautics Systems
Poway, CA
Anastacia.Macallister@ga-asi.com

INTRODUCTION

In 2021, The U.S. Intelligence Community was allocated a budget of about $85 million. Of that $85 million, about $23 million were allocated to the Military Intelligence Program (MIP), which supports tactical military operations (IC Budget, 2022). Many of these operations involve collecting large amounts of high-resolution video and imagery using satellites and unmanned aerial vehicles (UAVs). The MQ-9 Reaper, for example, uses a system known as Gorgon Stare, which is capable of producing 1.8 exabytes of high-definition video per day while imaging areas of 100 km^2 (Smith, 2013). As new video is collected, experienced Intelligence Analysts manually filter through data in order to detect and analyze military-relevant changes, then draw conclusions upon them over time. Such conclusions help drive foreign policy decisions and allow the Department of Defense (DoD) to adequately prepare for conflicts, perform battle damage assessments, assess troop buildups, and monitor foreign military equipment activity (Franz, 2016). Although existing DoD Intelligence Analysts are well trained, the volume and amount of data growth is rapidly outpacing their capacity to process it. An argument can be made that simply training more analysts to handle these data could solve this problem, however, the scale at which the data are growing is exponential in comparison to the pace at which analysts can be trained.

With so much data available, machine learning techniques could be employed to filter out militarily irrelevant imagery and as a result help alleviate the current analyst workload. This could allow analysts to focus on developing more informed and detailed analyses that would provide relevant decision aides to military personnel, such as pilots executing Intelligence, Surveillance and Reconnaissance (ISR) missions. Much of the current research in machine learning is focused on Deep Learning techniques (Galan, Carrasco, & LaTorre, 2022), which work to emulate a network of neurons, such as those found in the human brain. Using these techniques, it could be possible to ensure that all of the imagery collected by the military is analyzed without requiring significant resources. However, most of the DoD machine learning focus has been on performing Automatic Target Recognition (ATR), which is the ability to automatically classify objects in an image. While ATR may be useful in a real-time combat scenario, the models used fail to paint the whole battlespace picture adequately, because it doesn't take into account changes over time. Unfortunately, with the increase in data and geopolitical complexity, improved intelligence decision making requires capturing longitudinal trends between frames. In order to provide this longitudinal tracking, different types of neural networks need to be employed outside of the common ATR algorithms found in the DoD today.

Figure 1 shows the difference between common detection methods and those better suited to longitudinal change detection. Convolutional Neural Networks (CNNs), popular in ATR use cases, are used to provide classifications for objects using bounding boxes or other roughly located markers on the image (Baili, 2020). Algorithms for change or pattern of life detection, such as U-Net, provide a mask that shows the classification of each specific individual pixel (Ronneberger O., 2015). This pixel specific label is more accurate and can open the door to other post processing routines necessary for pattern of life analysis. In addition, due to the potential subtlety of changes in ISR imagery, using CNNs and their rough bounding boxes to perform change detection could produce more false positives and lead to overloading analysts with more information. However, by using an algorithm that provides a mask, techniques could be applied to threshold the changes based on desired criteria, and subsequently filter the changes that aren't considered militarily relevant. Unfortunately, research shows that

Figure 1. Comparison of Classification Methods

training any of these more detailed algorithms is heavily dependent on their application and use case. It is therefore necessary to prove out these concepts with surrogate datasets and incrementally build a system that can be re-trained on the appropriate data if they perform adequately.

The paper starts by describing background literature related to change and pattern of life detection. It then moves onto a description of the change detection architectures selected and how they performed using an unclassified and widely available dataset. Finally, the paper describes the design and initial implementation details of a Patterns of Life database for providing higher-order analysis of longitudinal changes in imagery. Ultimately, the paper details a prototype system for change detection, taking the first step towards developing a larger system that can be applied to DoD-relevant use cases.

BACKGROUND

The ever-increasing volume of data mentioned above is expected to flood analysis centers because of new and increasing sensing ability. Traditionally, humans or brittle deterministic based methods were used to help find objects of interest in images or video due to limits on human capital. However, these methods either are not scalable to the increasing amount of data or they are brittle and unreliable. As a result, interest in using machine learning and artificial intelligence techniques to help manage this data is growing. Deep learning, specifically, has shown great promise helping identify objects of interest in images and video from Automatic Target Recognition (ATR) applications to sentiment analysis in text. However, the types of approaches employed for these applications do not extend to other interesting data driven problems in the space, like change or pattern of life tracking/detection.

Traditionally, deep learning deployed in the DoD has focused on detecting the presence of an object of interest. This is categorized as an identification task. However, using this method, one cannot tightly localize the location of the object within the image due to the rough bounding box only approach (Louis & Sergeevna, 2018; Reisman, Dalrymple, Francisco, Konz, & Overman, 2021). This means that if a detection occurs, it's unclear where the object of interest stops and the world starts. This is a problem for change detection type tasks because changes cannot be identified. To really help offload processing and help make use of the massive amounts of data being collected, new methods need to be utilized to help detect localized changes in images and do pattern analysis to provide more higher-level insight to operators rather than just raw data or identifications.

Ronneberger, Fischer, & Brox looked at solving this detection and segmentation problem using a novel architecture called U-net (Ronneberger, Fischer, & Brox, 2015). Their work describes the need in medical imaging to not only know what something is but specifically where it is as well, for example, tumor boundaries for effective removal. They used an autoencoder type network that down filters images only to a subset of salient features for detection, much like traditional deep learning object detection network structures. However, once the network determines what is in the image, they up sample back to the full image resolution showing where the identified objects are at a pixel-by-pixel level as well. This provided the identification and localization required for more demanding computer vision type tasks like change and pattern of life detection. While this work was novel, it did not address the issue of detecting change in images over time or dealing with pattern of life considerations. Desantis et al. looked at using machine learning, specifically U-net based segmentation to detect building damage after natural disasters from satellite images (Desantis, Reisman, Konz, & Siddiq, 2020). They used a U-net to predict the amount of damage on a per pixel basis; from here they were able to extrapolate how much damage occurred in one area to help disaster response allocate the appropriate resources. Their work found that they could use threshold parameter tuning to help increase network accuracy but have trouble adequately assigning labels to pixels with low to moderate damage. This demonstrates the challenges associated with these methods and the tuning that can still be required. Francisco et al. looked at detecting changes in the location of ships in a port (Francisco, Reisman, Dalrymple, & LaTourette, 2021). They aimed to develop a pattern of life type of system that identifies when certain ships were present or not. They used U-nets to perform ship segmentation, then used classical deep learning to identify the ships. Then, they deterministically identified when those ships moved. They found their method to be accurate but that it required careful hand tuning and that it was sensitive to changes in images and use case.

Kostiantyn et al. looked at using the U-net architecture for deforestation detection in Ukrainian forests (Kostiantyn, Yushchuk, Khramtsov, & Seliverstov, 2021). They collected imagery from satellites and custom created the change maps to train the network. They trained six different network structures, varying parameters such as how the data was read into the network and if time series of images were provided to the network to make the change detection determination. They compared the F1 score and the dice coefficient to evaluate the performance of the models. They found that the models subtracting the first and second image in a time series, called the diff approach, without the extra time series information were the most accurate at detecting changes. They hypothesized this is because models using time series information have more parameters and are more complex to train, requiring a larger data set than often available.

Daudt et al. also looked at using different network structures to identify changes in high resolution satellite images using U-net derived architectures (Daudt, Le Saux, Boulch, & Gousseau, 2019). They also look to do land versus object categorizations. They found that their network also benefited from the diff approach for feeding in images over other more complex techniques. However, they cautioned that network structure is something that requires careful experimentation and fitting to the problem being solved. Lebedev et al. looked at using Generative Adversarial Networks (GANs) to address the issues surrounding generating data for change detection (Lebedev, Vizilter, Vygolov, Knyaz, & Rubis, 2018). They trained a GAN to generate images that could be used for training labeled change detection images. They had success generating limited realistic images for training, but still did not solve the issue surrounding high quality domain specific training data for change detection.

Peng, Zhang, & Guan looked at using a modified U-net architecture called U-net++ (Peng, Zhang, & Guan, 2019). This structure introduced skip level connections between layers of the network to help maintain localization info to help better segment out pixels of interest. They also pioneered using a modified loss function that weighted small classes to increase segmentation accuracy during training. They found that the introduction of skip connection greatly increases accuracy and specificity of segmentation. Daudt, Saux, & Boulch compared three different U-net architectures for end-to-end segmentation (Daudt, Saux, & Boulch, Fully Convolutional Siamese Networks for Change Detection, 2018). They introduced the U-net version called the Siamese U-net. This new structure uses a parallel image encoding method to allow for either subtracting images or adding them for change detection. They compared the traditional U-net with the additive or subtractive Siamese structure. They found that the diff method is the most accurate followed closely by the traditional U-net structure. This demonstrates that network structure matters and careful testing and development needs to be conducted for U-net architectures.

Work by Ji et al. looked to detect specific changes only in buildings using a U-net architecture (Ji, Shen, Lu, & Zhang, 2019). They designed a multi-model pipeline that first segments out buildings from an image then uses a change detecting U-net to detect changes. This work showed how different network structures can be designed for detecting patterns of life changes, however, they still did not address the larger database questions required to follow detection trends over time.

Overall, the work in the U-net space demonstrates the maturity of the method used for detecting changes in satellite imagery. However, literature points to the fact that careful experimentation and tuning of networks is required based on the specific use case. In addition, while literature addressed the detection portion of the work, little work is devoted to looking at how to build out a meaningful system to track and detect changes of interest in an analyst centric way. The paper looks at developing and tuning a U-net detection architecture for the specific use case of military change detection and begins addressing the challenges associated with tracking these changes in a meaningful contextualized way to provide pattern of life insight to analysts.

METHODS

There has been an extensive amount of work in the area of change detection using U-Net architecture variants with different features and advantages. However, in the case of change detection for pattern of life detection, it is important to select candidate architectures based on their features and performance metrics with regard to data sets created from satellite imagery. This section describes the original U-Net architecture, the U-Net++ and Siamese Nested U-Net architectures in detail, the ensemble method that was used to adapt them for our use cases, the training methods that were employed, a notional pipeline for use in an operational context and an example database architecture for logging changes. Additionally, an example instantiation of the pipeline is outlined.

U-Net Architecture

The network architecture for the original U-Net is composed of a contracting path and an expansive path. On the contracting side, two 3x3 convolutions are applied. After each convolution, rectified linear unit (ReLU) and 2x2 max pooling operations are applied. At this stage, down sampling occurs while doubling the number of feature channels. The expansive side begins with up sampling the feature map and then applying 2x2 convolutions to half the number of feature channels. Then, the feature map from the contracting path is concatenated, and two 3x3 convolutions with ReLU are applied. The final layer uses a 1x1 convolution to map the feature vectors to desired classes (See Figure 2. U-Net Architecture). While this architecture may work for some use cases, it falls short when fine-grained details in imagery are required to be captured such as for military chance detection in large overhead imagery.

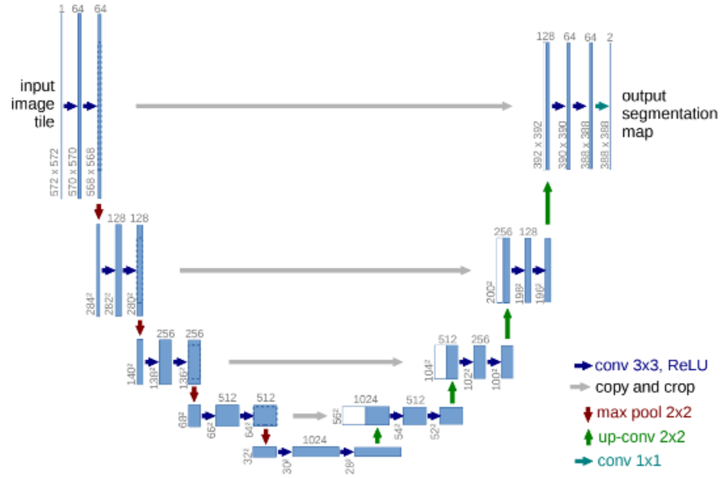

Figure 2. U-Net Architecture

Reprinted from U-Net: Convolutional Networks for Biomedical Image Segmentation, (2015), Retrieved from https://arxiv.org/pdf/1505.04597.pdf

U-Net++ Architecture

As previously mentioned, the U-Net++ architecture expands upon the original U-Net architecture by introducing skip connections (Zhou, 2018). Skip connections work by applying dense convolution blocks to feature maps. Each block's number of convolution layers is dependent on the pyramid level. Prior to each convolution layer, there is a concatenation layer that combines the output from the previous layer of the same block with the up-sampled output of the lower dense block that corresponds to it. This allows the semantic levels of the encoder feature maps to be closer to their corresponding feature maps in the decoder, which improves the performance of the optimizer. Formally, this process can be described as follows. Let $x^{i,j}$represent the output of node $X^{i,j}$ with i denoting the index of the down-sampling layer in the encoder and j denoting the index of the convolution layer with respect to the skip pathway. The stack of feature maps of $x^{i,j}$ can be expressed by:

$$x^{i,j} = \begin{cases} \mathcal{H}(x^{i-1,j}), & j = 0 \\ \mathcal{H}\left(\left[\left[x^{(i,k)}\right]_{k=0}^{j-1}, \quad (x^{i+1,j-1})\right]\right) & j > 0 \end{cases} \qquad (1)$$

where $\mathcal{H}(.)$ denotes a convolution followed by an activation function, $(.)$ denotes an up-sampling layer, and $[.]$ denotes the concatenation. Nodes at level $j = 0$ receive one input from the previous layer of the encoder, while nodes at $j = 1$ receive two inputs from the encoder at two consecutive levels, and nodes at $j > 1$ receive $j + 1$ inputs, where j inputs are the outputs of the previous j nodes in the same skip connection and the last input is the up-sampled output from the lower skip connection. Adding skip connections into the network structure allows fine-grained details to be detected in imagery; this can be important with regard to detecting militarily relevant changes such as changes in the operational posture of surface to air weapons.

Siamese Nested U-Net (SNUNet) Architecture

The SNUNet architecture, while still based on the original U-Net architecture, takes a slightly different approach. It uses a Siamese network as the encoder, where images that need to be compared are input into two branches and parameters are shared between them, guaranteeing that the same convolution filters are applied to extract features (S. Fang, 2022). Then, concatenation is used to combine the features between the two branches. Dense skip connections are used between the encoder and decoder to maintain high-resolution features and localization information. This process is described as follows. Let $x^{i,j}$represent the output of node $X^{i,j}$, which denotes a convolution block. The output $x^{i,j}$ can be expressed by:

$$x^{i,j} = \begin{cases} \mathcal{P}\left(\mathcal{H}(x^{i-1,j})\right) & j = 0 \\ \mathcal{H}\left(\left[x_A^{i,0}, x_B^{i,0},\ \mathcal{U}(x^{i+1,j-1})\right]\right) & j = 1 \\ \mathcal{H}\left(\left[x_A^{i,0}, x_B^{i,0},\quad [x^{i,k}]_{k=1}^{j-1}, \mathcal{U}(x^{i+1,j-1})\right]\right) & j > 1 \end{cases} \tag{2}$$

where $\mathcal{H}(.)$ denotes a convolution operation, $\mathcal{P}(.)$ denotes a 2 x 2 max pooling operation for down-sampling, $\mathcal{U}(.)$ denotes the up-sampling operation using transpose convolution, and $[.]$ denotes the concatenation operation. Nodes at level $j = 0$ cause the encoder to down-sample and extract features. Conversely, nodes at level > 0 cause the fine-grained features from the encoder to be shared with the decoder. In other words, this structure allows for detecting fine grained details in imagery while also maintaining accurate location information about those details; this is important because tactical decisions by military operators require precise location information.

Training Process & Ensemble Method

Designing a proof-of-concept system requires selecting the correct architectures, selecting a training dataset and selecting parameters that sufficiently demonstrate the efficacy of the architecture; this section will detail the process that was utilized for selecting and developing our proof-of-concept system. First, the architectures were each implemented in python. Then, each one was trained on a dataset of 256x256 pixel satellite imagery containing 10000 time-separated image pairs and binary change masks. Training was completed using a batch size of 8, 100 epochs, and a learning rate of 0.0001 for U-Net++, and a learning rate of 0.001 for SNUnet. The dataset utilized also includes 2998 time-separated image pairs and their binary change masks for validation. After training was completed, the models' outputs were compared. Given that both of the aforementioned architectures provide numerous benefits that appear to complement each other, a method to combine their outputs at inference time was devised. The ensemble method effectively uses image addition to aggregate the outputs of both change masks. There are conditions under which both algorithms produce different change detection data that is relevant, and in some cases, one of the algorithms misses the changes. Using the aggregation method allows all potentially relevant changes to be considered (See Figure 3. Example imagery with truth mask and output masks of selected models).

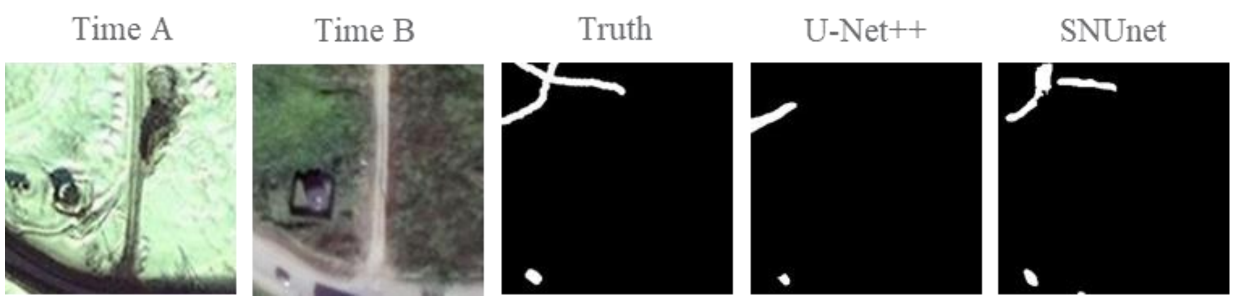

Figure 3. Example imagery with truth mask and output masks of selected models

Pipeline for Operational Concept

We propose a pipeline that utilizes the ensemble method and further expands upon it by integrating an object detection algorithm to identify the objects of interest in the corresponding imagery. Although the design and implementation of the object detection mechanism is out of the scope of this work, a minimum viable set of output data from the object detector was defined to facilitate the design and implementation of a change logging database. Figure 4. Full Pipeline Prototype describes the prototype pipeline design & implementation. The first step of the pipeline is to pass in the set of images, t_0 and t_1, to each model. Each model's change mask is then passed to the Aggregator, which performs the image aggregation to obtain a combined change mask. Then, the change mask is applied to t_0 and t_1 by the Extractor, thereby extracting the change pixels. The extracted change images are then passed to an Object Detector for inference, and the outputs of the object detector are logged to the database. After logging the changes, a status message is sent to the Analyzer. The Analyzer determines whether to query the database for any changes after the status message is

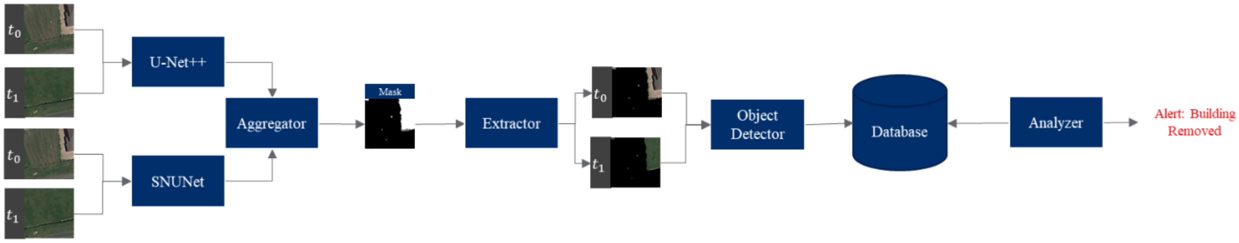

Figure 4. Full Pipeline Prototype

received and performs a high-level determination of the change in the imagery. In the case of the prototype system, the Analyzer is limited to a determination of whether an object was removed or added, and its identification. The database schema for the prototype system consists of a small amount of relevant metadata to support analysis at a high level, and will be expanded upon in future work (See Figure 5. Database Schema).

Consider an instantiation of this pipeline in a Command and Control (C2) center. Satellite imagery at t_0 is sent to the C2 center for processing. At t_1, the satellite acquires imagery for the same geographic location. The processing that occurs after the C2 center receives the imagery would include determining if the location has been imaged before, and if so, passing the images at t_0 and t_1 to the pipeline. After imagery pairs are processed in the background by the pipeline, identified changes are logged into the database and irrelevant data where changes of interest aren't detected is filtered out. At any time, an operator could designate an area of interest on a map-based display, which could then trigger a query on the database that would find and identify any changes between image pairs that fall within the geographic constraints. These changes could then be propagated to operators as alerts with context such that they could be verified and then forwarded to any assets that would benefit from increased situational awareness of the battlespace. Furthermore, as more data is logged into the database, long term trends and insights could be used to draw pattern of life conclusions.

Overall, the U-net based architectures of U-net++ and SNUnet, which were selected for use in the proof-of-concept system, have features that complement each other. The work described outlines a proposed method for getting the best of both worlds by using a pipeline that performs a significant number of tasks that military analysts are currently responsible for. Integrating a system such as the one outlined could save a lot of time for analysts and therefore enable the development of more comprehensive data analytics algorithms to gain a more complete understanding of the battlespace. While this work outlines a potential sequence of events for using the pipeline, further work will be required to support other militarily relevant use cases.

Field	Type
ID	int
Type	text
Confidence	double
Latitude	double
Longitude	double
Time Delta	double
Capture Type	text
Change Type	text
Pixel Coordinates (x_0, y_0)	int, int
Pixel Coordinates (x_1, y_1)	int, int
Image ID t_0	int
Image ID t_1	int
Image Path t_0	text
Image Path t_1	text

Figure 5. Database Schema

RESULTS

The challenge with the DoD's intelligence community lies within the ability to quickly process and analyze terabytes of imagery data every day. The human operators are struggling to keep up with the increasing volume while maintaining the status quo for detecting changes in the data which can be critical for early warning on emerging threats. This paper attempts to offer a unique solution for processing large amounts of data quickly using artificial intelligence and computer vision to identify changes in imagery. This solution could potentially save operators countless hours of manually looking through footage and allow them to focus their attention on the emerging threats and supporting the warfighter. This section will walk through the testing of both the U-net++ and the SNUnet on a set of publicly accessible satellite imagery datasets. This section will detail the performance and the pros & cons of both algorithms.

Data is the foundation for all analytical approaches, and it deserves keen attention when evaluating model performance. The data set garnered for this experiment is a publicly accessible satellite imagery set of high-resolution images of the earth. This data set includes two sets of images where one set has fourteen 4725x2700 pixel images and the second set has four 1900x1000 pixel images, and each set has images of the same geographical locations except in different points in time. Since high resolution models are often extremely noisy and require a lot of memory to process, the images were broken into many 256x256 pixel images. This image chopping method is valid and preferable because the objects being detected for change represent a small portion of each original image. The chopped images are large enough to contain the changes while also reducing noise and other unique features in the images that might unintentionally influence the model. Following the transformations, the data set is then further split into the standard training, testing, and validation sets which are used to train and evaluate the models. Now that the data has been described and staged, it is important to understand what the models are attempting to do in order to properly evaluate their performance.

Analysts spend enormous amounts of time reviewing satellite images for changes in patterns of life. These modeling techniques are meant to alleviate the need for analysts to review all the images and focus their attention on key changes within the images. This means that well performing models only flag meaningful changes instead of all changes. These meaningful changes include additions or removals of roadways, buildings, vehicles, and other indications of human activity. The models are attempting to predict which pixels in every image represents a change compared to the previous image. This means that the model makes 65,536 binary predictions on each image. The binary mask output is compared to the truth binary mask in order to create the classification confusion matrix where true positives are pixels with change and correctly predicted as change, false positives are those with no change but false predicted as change, true negatives are pixels with no change and correctly predicted as no change, and false negatives are pixels with change but falsely predicted as no change. This method allows us to generate classification performance metrics such as precision, accuracy, recall, and F-1 score for each image. This method allows us to compare the different models' performance distributions across the 2,942 test images and the 2,998 validation images.

This experiment analyzes the prediction accuracy for the SNUnet, U-net++, and the ensemble models. Figure 6 displays the distribution of each image pixel prediction F-1 scores. This analysis method allows us to directly compare the prediction performance of each model over many images. This experiment's results show that the U-net++ model greatly outperforms the SNUnet model while the ensemble performs just slightly worse than the U-net++. The results show that the median F-1 score the ensemble and the U-net++ are both around 0.7 which demonstrates a reasonable prediction performance whereas the SNUnet has a median score around 0.4 mark. Additionally, the F-1 distribution of the SNUnet model is much larger than the other two which is driven by the SNUnet model not predicting any pixel

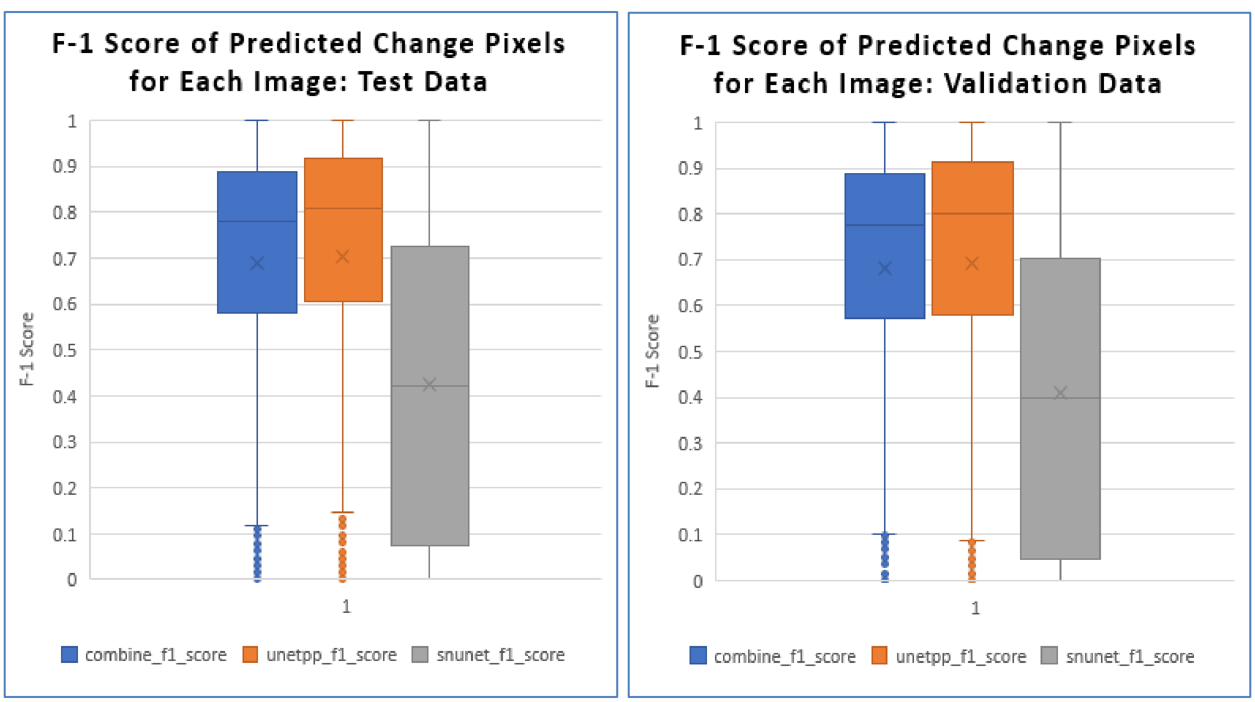

Figure 6: F-1 Scores for Pixel Prediction where each point is an image

changes in 7.6% of the images. This is likely due to the effect that the Siamese network is heavily influenced by its hyper parameters, and further tuning of these parameters could lead to drastically increasing its performance.

The more important result of this experiment is the reliability of the combined ensemble for this problem. Figure 7 shows two scatterplots of the F-1 scores for both the U-net++ and the SNUnet as compared to the combined ensemble method. The points above the diagonal line mean that the individual model outperformed the combined ensemble for those images, whereas the points below the diagonal mean represent the combined ensemble outperforming the individual model. The SNUnet model showed nearly all of its predictions were worse than the combined ensemble. Whereas the U-net++ model shows an equal or slightly greater values to the combined ensemble. This is significant because the combined ensemble retained high performance even though it has a worse performing model influencing its output. The results are an important attribute of ensemble methods as they are less influenced by any one model. This resiliency means that the models that perform well on some data will make up for the other models which perform worse on the same data and vice versa. The combined ensemble demonstrates that it can retain its high-performance while being more resilient with varying data sets. For example, if one model is biased towards identifying change in buildings while another model is biased towards roads, the combined ensemble would be able to find both with a high confidence. Additionally, the combined ensemble would be able to negate the effects of one model performing poorly where another performs well and vice versa. This resiliency builds operator trust in the solution and provides them the ability to focus on more crucial tasks.

Figure 7: Comparison of F-1 scores between the Combined ensemble model and the other two models

While this technology and approach is still emerging, the experiment's results show great potential in their ability to save analysts time from manually reviewing many individual images for change. However, this technology is still being matured through further experimentation to expand upon the bounded problem described in this paper. Some of these expansion factors include increasing the number of images for training, testing, and validation, further hyper parameter tuning, adding more models to the ensemble, including more types of changes, varying the image sizes, and including synthetic data (e.g. darkening, lightening, blurring) to make the model more robust. These factors, which are major influencers to this current experiment, will be addressed in future efforts as this technology is further refined. The technology does show promise for its ability to automatically scan thousands of images and identify the changes within the images to save analysts countless hours of manual processing. This in turn, will help enable the DoD to redirect resources to important decision-making tasks in support of the warfighters.

CONCLUSION AND FUTURE WORK

With increasing imagery data availability, organizations such as the DoD can benefit from machine learning techniques to perform Patterns of Life analysis tasks such as change detection more efficiently. While most of the current applications of machine learning in the DoD have focused on Automatic Target Recognition tasks, tracking changes over longer periods of time allows for a more comprehensive understanding of the battlespace. While current processes to perform imagery analysis tasks exist, they are not reliable in terms of consistently being able to ingest and process all the data. Additionally, these processes are not scalable given the growth of the amount of data; eventually, the amount of data will outpace the availability and capacity of analysts to process it.

The work presented in this paper describes our implementation of a prototype system that performs change detection as a first step to developing a larger system that can be applied to DoD-relevant use cases. An ensemble method that combines the outputs of two U-Net based architectures was developed. Additionally, a pipeline for the operational concept was illustrated, and an example database schema with relevant data was created. Finally, an evaluation of the ensemble performance in comparison to the performance of the individual models was completed. The evaluation showed that the implementation of a combined ensemble model provides resiliency and accuracy benefits in contrast with only using one model. Moving forward, future work will continue to build upon the proposed pipeline by focusing on development of higher-order analysis algorithms that can provide insights from data stored in the database.

REFERENCES

Baili, N. (2020). Automatic target recognition with convolutional neural networks. *Electronic Theses and Dissertations*, Paper 3670. Récupéré sur https://ir.library.louisville.edu/etd/3670

Daudt, R., Le Saux, B., Boulch, A., & Gousseau, Y. (2019). Computer Vision and Image Understanding Multitask learning for large-scale semantic change detection.

Daudt, R., Saux, B., & Boulch, A. (2018). Fully Convolutional Siamese Networks for Change Detection.

Desantis, Z., Reisman, M., Konz, L., & Siddiq, S. (2020). Broad area damage assessment. *Proceedings - Applied Imagery Pattern Recognition Workshop.* Institute of Electrical and Electronics Engineers Inc.

Francisco, A., Reisman, M., Dalrymple, J., & LaTourette, K. (2021). Deep learning-based object level change detection in overhead imagery. *SPIE-Intl Soc Optical Eng.*

Franz, N. (2016). Targeted killing and pattern-of-life analysis: weaponised media. *Media, Culture & Society*, 39.

Galan, J. J., Carrasco, R. A., & LaTorre, A. (2022). Military Applications of Machine Learning: A Bibliometric Perspective. *Mathematics, 10*, 1397.

IC Budget. (2022). Récupéré sur Office of the Director of National Intelligence: https://www.dni.gov/index.php/what-we-do/ic-budget

Ji, S., Shen, Y., Lu, M., & Zhang, Y. (2019). Building instance change detection from large-scale aerial images using convolutional neural networks and simulated samples. *Remote Sensing*.

Kostiantyn, I., Yushchuk, M., Khramtsov, V., & Seliverstov, O. (2021). Deep Learning for Regular Change Detection in Ukrainian Forest Ecosystem with Sentinel-2. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Lebedev, M., Vizilter, Y., Vygolov, O., Knyaz, V., & Rubis, A. (2018). Change detection in remote sensing images using conditional adversarial networks. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives.*

Louis, K., & Sergeevna, A. (2018). Unsupervised Change Detection in Satellite Images Using Convolutional Neural Networks.

Peng, D., Zhang, Y., & Guan, H. (2019). End-to-end change detection for high resolution satellite images using improved UNet++. *Remote Sesnsing*.

Reisman, M., Dalrymple, J., Francisco, A., Konz, L., & Overman, T. (2021). Closely spaced object segmentation using a hybrid deep learning approach. *SPIE-Intl Soc Optical Eng.*

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention*.

S. Fang, K. L. (2022). SNUNet-CD: A Densely Connected Siamese Network for Change Detection of VHR Images. *IEEE Geoscience and Remote Sensing Letters, 19*, 1-5. doi:10.1109/LGRS.2021.3056416

Smith, M. (2013, Jan 29). *DARPA's unblinking, all-seeing 1.8-gigapixel camera stare on PBS Rise of the Drones*. Récupéré sur CSO: https://www.csoonline.com/article/2223941/darpa-s-unblinking--all-seeing-1-8-gigapixel-camera-stare-on-pbs-rise-of-the-drones.html

Terabytes From Space: Satellite Imaging is Filling Data Centers. (2020, April 28). Récupéré sur Data Center Frontier: https://datacenterfrontier.com/terabytes-from-space-satellite-imaging-is-filling-data-centers/#:~:text=Commercial%20imaging%20companies%20are%20collecting,measured%20in%20petabytes%20per%20company.

Zhou, Z. R. (2018). Unet++: A nested u-net architecture for medical image segmentation. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support - 4th International Workshop,* *DLMIA 2018 and 8th International Workshop, ML-CDS 2018 Held in Conjunction with MICCAI 2018*, 3-11.