A Comprehensive Summary of Three Years of Research Contributions in Computer Vision, Machine Learning, and Beyond

Matthew Ciolino[1]

[1]PeopleTec, Inc
[1]mrciolino@alum.lehigh.edu

***Abstract*—We have made contributions in the fields of computer vision, pattern recognition, machine learning, artificial intelligence, computation and language, cryptography and security, and software engineering. Our works have focused on various topics such as rapid object detection, sub-typing, text augmentation, image compression, super resolution, and adversarial risk to machine learning models. Our research has also included automating defense against adversarial attacks and discovering model characteristics through strategic probing. Additionally, we have applied our expertise to the gaming world by developing generative language models for chess and go gameplay. Our papers have been published in various formats and conferences, including SPIE, AI4I, NIAI, ICAIT, and NFCS. This is what we did.**

***Index Terms*—Computer Vision and Pattern Recognition, Machine Learning, Artificial Intelligence, Computation and Language, Image and Video Processing, Cryptography and Security, Software Engineering, Adversarial Risk, Adversarial Attacks, Neglected Languages, Generative Language Models, Game Play, Strategic Probing, White Box Model Characteristics**

Introduction

The following work represents a team's efforts on various research questions. I will mention the other authors here and recognize their efforts: David Noever, Josh Kalin, Dominick Hambrick, Willie Maddox, Grant Rosario and Wes Regian. Thank you all for your intellectual conversations and commentary over the years on these papers and on the Machine Learning industry as a whole. The following paper represents a subset of our efforts over the past three years to the following subjects [Table I]

> The following summaries are initially generated with ChatGPT [1] based on the contents of each paper. It was asked to return the goal, experiment, and result. That text was reviewed/edited and then images from each paper were selected to add.

I. Discoverability in Satellite Imagery: A Good Sentence is Worth a Thousand Pictures [2]

In our paper "Discoverability in Satellite Imagery: A Good Sentence is Worth a Thousand Pictures," we aimed to evaluate the effectiveness of various machine learning models for generating captions for satellite images [Figure 1]. We compared seven models on the largest benchmark for satellite image captions, extending the labeled image samples five-fold, and augmenting, correcting, and pruning the vocabulary. The results showed that the best satellite image model for classification was NASNetMobile, which was also the smallest in model size, but the best captioning outcome was the larger VGG19 model.

TABLE I: Paper Subject Counts on arXiv

Subject	Subject Count
Machine Learning (cs.LG)	10
Computer Vision and Pattern Recognition (cs.CV)	7
Computation and Language (cs.CL)	5
Cryptography and Security (cs.CR)	4
Artificial Intelligence (cs.AI)	2
Image and Video Processing (eess.IV)	2
Software Engineering (cs.SE)	1
Computer Science and Game Theory (cs.GT)	1
TOTAL PAPERS	16

Fig. 1: xView Image to Caption (Trained on RSICD)

We also introduced a novel multi-class confusion or error matrix to score both human-labeled test data and never-labeled images. The results were compared to a previous study by Lu, et al. (2017), and the BLEU scores shown in our paper exceeded the best sequences for multimodal methods on RSICD. Our study suggests that the reduced NASNetMobile model offers a faster approach in resource-scarce environments typically expected for edge computers.

In conclusion, our study demonstrates that smaller image

models can provide accurate results without sacrificing overall accuracy and offers new deployment opportunities for satellite image captioning. Our results suggest future captioning strategies that can enrich the class coverage beyond land use applications and lessen color-centered and adjacency adjectives.

II. Super Resolution Effect on Object Detection and Image Classification [3]

The goal of the paper is to study the effect of different training sets on the performance of Single Image Super Resolution (SISR) [Figure 2] using the Super Resolution Generative Adversarial Network (SRGAN). We aim to explore how diverse training sets and the presence of objects in the test ontology impact the SISR performance and its downstream applications in computer vision tasks, such as binary classification and object detection.

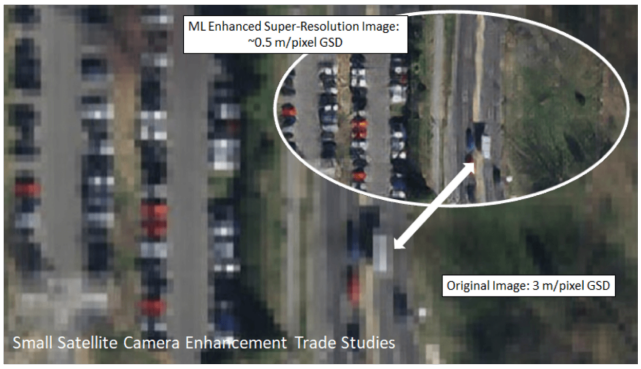

Fig. 2: Super Resolution on Planet Labs' SkySat Imagery

We conducted experiments on two datasets: Skysat's 0.8m samples for training and Planet's 3.0m Shipsnet images for testing. They trained five SRGANs on different land-use classes (agriculture, cities, ports, etc.) and evaluated their SISR, binary classification, and object detection performance on the same unseen dataset. The results of these experiments provide insights into the relationship between training sets, SISR, and downstream computer vision tasks.

The results of the experiments showed that curated training sets that contain objects in the test ontology perform better on computer vision tasks. The super resolution models with diverse datasets and objects in the test ontology had better results in object detection, with an average improvement of 18.4% in mAP compared to the raw image trained model. On the other hand, image classification showed marginal improvement, which depends on the difficulty of the test set. The conclusion of the paper is that having a diverse dataset with objects in the test ontology allows for better performance of downstream tasks after super resolution.

III. Black Box to White Box: Discover Model Characteristics Based on Strategic Probing [4]

We present a method for classifying classifiers in machine learning. Our approach involves using a deep classifier trained on input probes and the outputs of common machine learning models. We explore two subdomains in machine learning: image-based classifiers and text transformers with GPT-2. Our experiments aim to determine the underlying architecture and primary training dataset of a machine learning model.

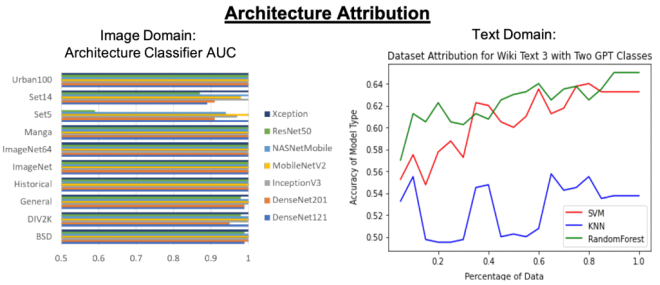

Fig. 3: Architecture Attribution Experiments on Image and Text Architectures

In our evaluation section, we conduct experiments on image and natural language processing. We use publicly available datasets and pre-trained models from popular libraries such as Keras and HuggingFace. We demonstrate the ability of our classifier to determine the dataset type from a single image and the architecture of a text transformer from its outputs [Figure 3].

Our contribution to the field of adversarial attacks lies in offering a process for discovering underlying model information for use in adversarial attacks. This is important because prior work in adversarial attacks often assumes that the attacker has knowledge of the model's architecture and training dataset, but our paper provides a method for obtaining that information. The results of our experiments show that our proposed method is effective in determining the architecture and dataset of a machine learning model.

IV. The Go Transformer: Natural Language Modeling for Game Play [5]

We explored the application of language transformers in playing the ancient game of Go. The goal was to generate plausible strategic moves in the game, by fine-tuning the Generative Pretrained Transformer (GPT-2) with the style of Go champions as archived in the Smart Game Format (SGF). This was achieved by training the model to mimic the text descriptions of move sequences in SGF. The fine-tuned GPT-2 was then able to generate valid, previously unseen strategies for Go [Figure 4], by preserving the punctuation and spacing in the raw output of the text generator.

In the experiment, we compared the performance of the fine-tuned GPT-2 with random game boards. The results showed that the language model was able to capture both the sequencing format of championship Go games and their strategic formations. The fine-tuning process resulted in efficient opening move sequences that favored corner play over center and side play, which was a significant improvement over random game boards.

The results of the experiment demonstrate that language modeling can be a useful tool for generating strategies in

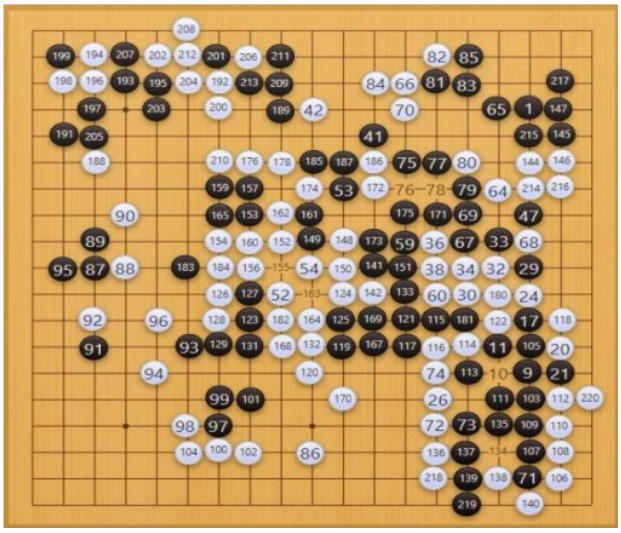

Fig. 4: Medium GPT-2 Go Transformer play after 41k training steps. White leads in both Area and Territory, by 31 and 27 respectively. Number indicates play order.

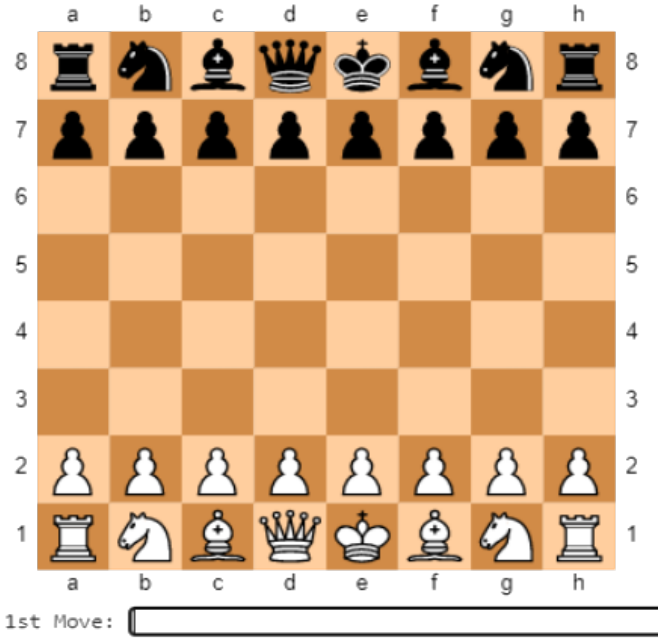

Fig. 5: Human vs. Machine in Live Play with Chess Transformer. The Colaboratory notebook includes pre-trained medium GPT-2 models and instructions for human (white) vs. machine (black) in head-to-head game play.

board games, and that this approach offers novel opportunities for more than 40 other board games where historical text annotations provide training data. Game generation as a language modeling task provides a new direction for the study of game playing algorithms, both in language and gameplaying. By applying these techniques, we hope to contribute to the advancement of machine learning algorithms for solving high-dimensional games, and to provide additional inputs for training and generating algorithms in language, imagery, and audio.

V. The Chess Transformer: Mastering Play using Generative Language Models [6]

We present the Chess Transformer, a language model that generates meaningful moves on a chessboard by training on 2.8 million chess games in Portable Game Notation (PGN) format using OpenAI's Generative Pre-trained Transformer (GPT-2). The transformer's architecture features built-in parallelism and a directed attention mechanism, allowing it to generate plausible strategies that are identifiable as classic openings. Our research demonstrates that language transformers can support more generic strategic modeling, particularly for text-archived games.

The fine-tuning of the GPT-2 transformer with 774 million parameters resulted in the generation of plausible chess moves. Our contribution to the field of chess modeling also includes a novel game interface where human players can challenge the transformer in live play, hosted on Google's Colaboratory platform [Figure 5]. In contrast to previous work on chess models, our approach is language-based and does not rely on decision trees or Monte Carlo Tree Search. Instead, the transformer's architecture features encoder-decoder cycles that apply weights to features, with its unique attention mechanism effectively overweighing the most relevant features as it learns.

The application of language models like GPT-2 to chess highlights new ways to expand chess training data with high-level simulation data. Our research contributes to the growing interest in cross-domain applications of models outside their initial language-related training sets. The remarkable abilities of transformers to generate text arise from its parallelism during training, which allows it to ingest vast amounts of internet-scale textual inputs. We anticipate that future work will build on the promise of the Chess Transformer, particularly in other strategy games where features can capture the underlying complex rule syntax from player annotations.

VI. Local Translation Services for Neglected Languages [7]

As a researcher in the field of computer translation, I, along with my team, have explored the possibility of using lightweight, high-quality translators for low-resource and less popular languages. This study focuses on translating two historically interesting, but obfuscated languages: hacker-speak ("l33t") [Table II] and reverse (or "mirror") writing as practiced by Leonardo da Vinci. Our main contribution is to generalize a deep learning architecture to translatable variants of hacker-speak, including lite, medium, and hard vocabularies.

The work involves training a long short-term memory recurrent neural network (LSTM-RNN) on bilingual sentences to support deep learning models. Our results show that the LSTM recurrent neural networks translate complex bilingual pairs with lightweight models for 26 additional (non-obfuscated) languages. The best-performing models are ranked in order,

TABLE II: Bilingual Evaluation Understudy (BLEU) scores for Translator Per Bilingual Pairs in the Obfuscated Examples

Language	English	Mirror	Leet Lite	Leet Mid	Leet Hard
Translation	She just left	tfel tsuj ehS	Zh3 juz7 l3f7	5aych3 _—M57 13ph7	esaych3 JY3Wes1 lJ3ph7
BLEU-1	-	0.95	0.63	0.37	0.23
BLEU-2	-	0.93	0.55	0.25	0.11
BLEU-3	-	0.9	0.46	0.14	0.26
BLEU-4	-	0.72	0.22	0.02	0.32
Interpretation	-	May Exceed Human	Fluent	Understandable	Gist is clear

with Italian as the most successful and Mandarin Chinese as the most challenging. The results of the translations are evaluated using the Bilingual Evaluation Understudy (BLEU) scores, which evaluate the closeness of the translator's output on unseen test data compared to a set of high-quality reference translations.

The BLEU-4 score penalizes missed word substitutions and ordering up to 4 tokens in a sentence sequence compared to the reference translation. Our results show that the LSTM works symmetrically by swapping columns and the English-to-Italian vs. Italian-to-English models work as a level between "expert" to "fluent" in reverse inputs. The BLEU scores indicate that our models achieve a range between understandable to better-than-human translations. These results demonstrate the potential of using lightweight translators for neglected languages and lay the foundation for future work in translating technical (medical or legal) jargon, processing health records, and handling many other dialects.

VII. Automating Defense Against Adversarial Attacks: Discovery of Vulnerabilities and Application of Multi-INT Imagery to Protect Deployed Models [8]

The goal of this paper is to defend machine learning models from adversarial attacks in image classification. Adversarial attacks can manipulate or evade the detection of a machine learning model by altering the features used by the classifier. The proposed solution is to use Multi-INT imagery, specifically 3-color channels plus infrared for vehicles, in combination with ensemble learners.

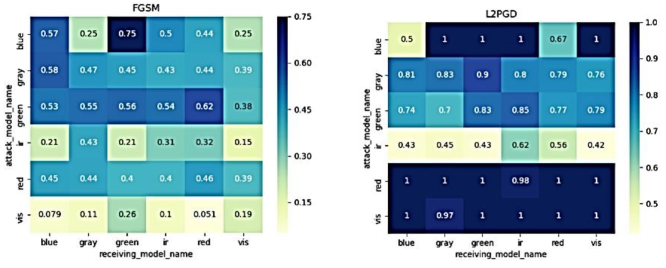

Fig. 6: Adversarial Surface for MobileNetV2 Models trained on VEDAI dataset

In the experiment, we evaluated the use of Multi-INT imagery to combat adversarial attacks using the Vehicle Detection in Aerial Imagery (VEDAI) dataset. The dataset includes multiple channels of data, including visible, infrared, and gray images of vehicles. The authors trained a MobileNetV2 model on the VEDAI dataset and then subjected the model to white-box attacks using the FoolBox library. The FoolBox library was used to generate six types of attacks, including Fast Gradient Sign Method, Fast Gradient Method, Projected Gradient Descent, and L2DeepFool.

The results of the experiment [Figure 6] showed that the use of Multi-INT imagery was effective in defending the model from adversarial attacks. The combination of visible and infrared channels, in particular, improved the robustness of the model against attacks, uncovering vulnerabilities and correcting them with supplemental data inputs. These results demonstrate the potential of combining offensive and defensive techniques, in rough analogy to the idea of a "green team," to protect machine learning models from adversarial attacks in overhead applications.

VIII. Fortify Machine Learning Production Systems: Detect and Classify Adversarial Attacks [9]

We aim to fortify machine learning production systems by detecting and classifying adversarial attacks. Adversarial actors are constantly attacking these systems, making it crucial for deep learning models to accurately detect fake or adversarial input while maintaining speed. In this paper, we propose a solution to detect the incoming adversarial attack and its characteristics, which will help to train the underlying model in a structured manner to be robust from those attacks and also potentially filter out the attacks in real-time.

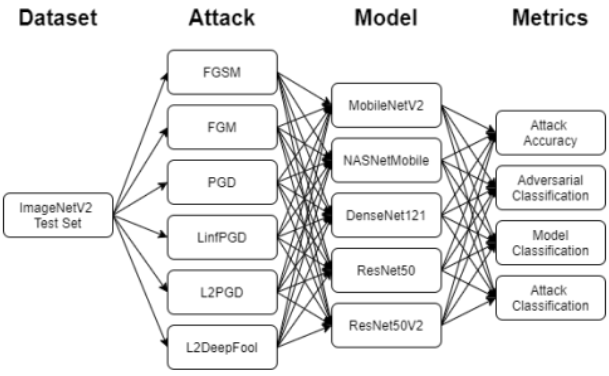

Fig. 7: Adversarial Image Experiment with the Attack Type, Model Architecture and Metrics used

We start by fine-tuning MobileNetV2 on different 50-class datasets and run inference from those models on a mix of super resolution datasets. By collecting the model outputs, we train a classifier to predict which dataset MobileNetV2 was trained on based on a single model output, achieving 0.97 average precision. Then, we extend this work to detect adversarial attacks. Our goal is to detect if a model was attacked, the dataset that was used to train the model, the model used, and the attack used, all from a single model output.

Our approach [Figure 7] to detect adversarial attacks is novel compared to previous methods as we attempt to create a universal detector for adversarial attacks and their characteristics. Our experiments demonstrate the defense against adversarial attacks on image classification models. We show that our detector can classify attacks in real-time and with high accuracy. Our results highlight the need for production machine learning systems to detect and classify adversarial attacks, and our approach provides a solution to meet that need.

IX. A Modified Drake Equation for Assessing Adversarial Risk to Machine Learning Models [10]

We present a modified version of the Drake Equation to assess the risk of adversarial attacks on machine learning models. Adversarial attacks pose a significant risk when deploying machine learning models in production, and our goal is to provide a semi-quantitative benchmark for evaluating and estimating the potential risk factors. Our modified Drake Equation takes into account several factors that influence the risk of adversarial attacks, such as the popularity of the model, the size of the sponsoring enterprise, and the monoculture of adoption [Figure 8].

$$N = R * fp * ne * fl * fi * fc * L$$

N = *the number of successful adversarial attacks*
R = *average enterprise size*
fp = *fraction of models published, named, open sourced or fielded in the wild*
ne = *average number of* **e***ngineered parameters (memory, billions of parameters)*
fl = *fraction of* **l***earning ratio, as training/test data or active hybrid feedback*
fi = *fraction of* **i***nput supervisory and quality control steps*
fc = *fraction of* **c***ompleted queries that return detectable or logged answers*
L = **l***ength of time that attackers can query without consequences or timeout*

Fig. 8: Summary of Components of the Modified Drake Equation

We conducted an experiment using six popular models as examples and estimated their adversarial risk using our modified Drake Equation. The results indicate that newer, larger architectures are less vulnerable to adversarial attacks compared to older models. However, there is room for improvement in these experiments, such as exploring the properties of specific model architectures in greater detail.

We also conducted a correlation analysis to understand the dependencies between the factors and the adversarial risk. The results reveal that the fraction of the learning ratio is highly correlated with adversarial risk, as well as with the number of parameters. Additionally, the fraction of input supervisory guidance is correlated with the fraction published, and the fraction completed queries is highly correlated with the fraction published. These findings provide insight into the relationships between the factors and adversarial risk, helping us to better understand the potential risk of deploying machine learning models.

X. Color Teams for Machine Learning Development [11]

In this paper, we propose a new teaming construct for machine learning development teams to better defend against adversarial attackers. The color teams approach, inspired by the InfoSec Color Wheel in cybersecurity, assigns clear responsibility to individuals for the baseline (Yellow), attack (Red), and defense (Blue) aspects of the pipeline. Combining colors leads to additional knowledge sharing and more robust models. The responsibilities of the new teams Orange, Green, and Purple will be outlined in the paper.

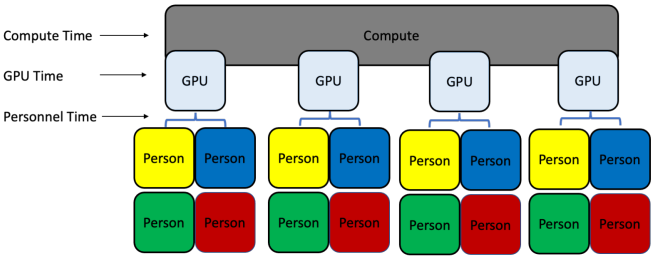

Fig. 9: Summary of key resource allocation steps to move a project from conception, training, testing and deployment with their assigned color team

We aim to apply the concept of the color wheel [Figure 9] to machine learning development, in order to build more robust models from the beginning of the process. By including attackers in the development loop, the entire team can better understand the vulnerabilities of the system. The Yellow Team, also known as the Development Team, is responsible for building the machine learning model. The Red Team serves as the attackers, understanding known vulnerabilities and exploitative methods. The Blue Team, on the other hand, defends against the Red Team's attacks and provides solutions for the Yellow Team to implement.

Each team in the classic configuration has clear responsibilities, and blending the teams into mixtures of colors has become necessary due to a lack of communication and understanding of vulnerabilities in common development platforms. In future work, we will map these additional team constructs into the machine learning development lifecycle. The theoretical rate-limiting step is the Yellow Team which builds the initial code, and we aim to improve the efficiency of the entire pipeline by combining the strengths of each team.

XI. Enhancing Satellite Imagery using Deep Learning for the Sensor To Shooter Timeline [12]

As researchers at PeopleTec Inc, we aimed to improve the sensor to shooter timeline in satellite imagery. The timeline is typically affected by two main variables: satellite positioning and asset positioning. We investigated two main ideas in this paper: increasing the effectiveness of satellite imagery through image manipulation and the effect of on-board image manipulation on the sensor to shooter timeline.

To accomplish this, we explored four scenarios through Discrete Event Simulation (DES) [Figure 10]: comparing on-board processing and ground station processing, cloud cover removal, super resolution, and image to caption. Our research showed that image manipulation techniques, such as super resolution, cloud removal, and image to caption, can significantly

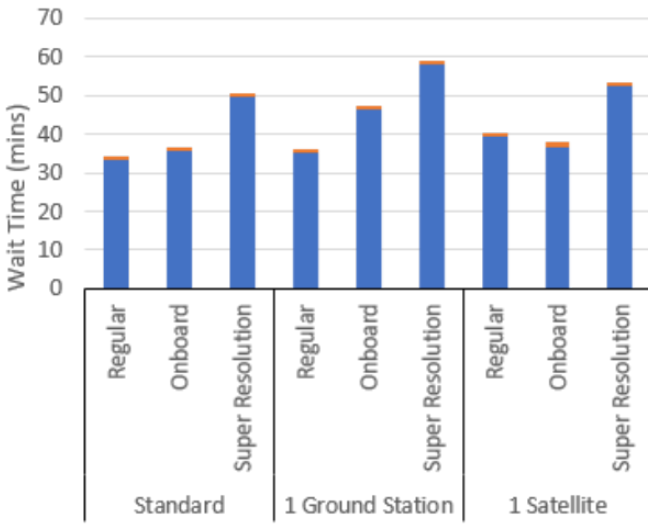

Fig. 10: Discrete Event Simulation Results with various timings for processing, transmission, and location

improve the quality of delivered information and thus improve the sensor to shooter timeline.

In conclusion, our findings showed that increasing the effectiveness of satellite imagery through image manipulation can greatly improve the sensor to shooter timeline. The use of super resolution, cloud removal, and image to caption improved the quality of information delivered, making the process quicker and more efficient. By considering the man in the loop process and implementing these techniques, the ISR community can greatly improve the sensor to shooter timeline in satellite imagery.

XII. Image Compression and Actionable Intelligence With Deep Neural Networks [13]

The goal of this paper is to investigate information reduction techniques that can be used to deliver satellite imagery information to disadvantaged users who are working on low-connectivity devices. The paper proposes a survey of these techniques to find a way to deliver the information in a smaller package. The four techniques explored are traditional image compression, neural network image compression [Figure 11], object detection image cutout, and image to caption.

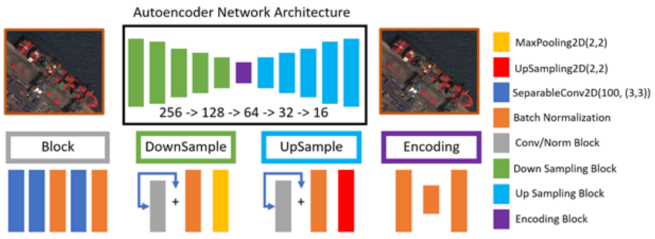

Fig. 11: Autoencoder Architecture for Alternative Image Compression Methods

The experiment was designed to rank each method according to its impact versus data size ratio. The methods were compared based on common image metrics such as peak-signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). The authors also considered alternative forms of data compression, including image to caption and object detection. These methods are a combination of computer vision models and natural language processing models, and they aim to compress the information from an image into a sentence that can be used to describe it.

The results showed that neural network image compression, also known as autoencoding, was the best method for compressing satellite imagery. This is a lossy compression method where an image is compressed to a representative state (embedding) and then expanded back into an image. We used the compression ratio and PSNR were used to rank the autoencoder compression models. We suggest using a tiered response system where the smallest data source is sent first following up with more information (i.e. image to caption, then autoencoder embedding, then image cutout, and finally the raw image)

XIII. Back Translation Survey for Improving Text Augmentation [14]

We (the authors) aim to investigate the effect of 108 different language back translations on various Natural Language Processing (NLP) metrics and text embeddings. Back translation is a text augmentation technique that translates text from English to another language and then back to English. The goal of this study is to understand the full effect of back translation on NLP models and determine which language back translations might be a better candidate for improving text augmentation.

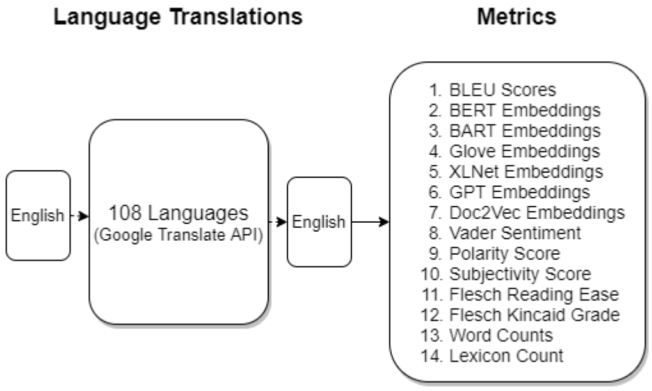

Fig. 12: Experimental flow diagram showing the languages used for translations followed by the metrics used to analysis the differences

To achieve this goal, we conducted an experiment [Figure 12] by translating 1000 random English tweets from the Sentiment-140 dataset using the Google Translate API. We then translated the tweets to all 108 languages supported by Google Translate and back to English. After the back translation process, we analyzed the differences using various text metrics, such as Bilingual Evaluation Understudy Score

(BLEU), BERT embedding distance, BART embedding distance, GPT embedding distance, XLNet embedding distance, GloVe embedding distance, and Doc2Vec embedding distance.

The results showed that back translations from some languages had a larger affect on NLP metrics while back translations from other languages had a smaller or null effect. For example, back translations from German, Spanish, and French showed smaller effects on NLP metrics, while back translations from some Asian languages showed larger effects. In conclusion, we found that the effect of back translation varies depending on the language and text embedding used, and future research should focus on finding a way to determine the optimal language for back translation to improve text augmentation.

XIV. The Turing Deception [15]

In our research, we revisited the classic Turing test and evaluated the abilities of recent large language models, such as ChatGPT, to imitate human-level comprehension and generate compelling text. Our experiment consisted of two task challenges: summary and question answering, in which ChatGPT was prompted to produce original content from a single text entry and sequential questions initially posed by Turing in 1950.

	Real	Fake	Tokens
Turing Original	99.9	0.1	295
ChatGPT App. A	0.02	99.98	314
ChatGPT App. B	0.02	99.98	236
ChatGPT App. D	0.32	99.68	497
ChatGPT App. E	0.02	99.98	384
ChatGPT App. F	99.98	0.02	422
Paper Intro	92.42	7.58	376

Fig. 13: OpenAI Detector Scores based on GPT-2 Applied to ChatGPT output and Turing

To measure the originality and undetectability [Figure 13] of the generated content, we scored it against the OpenAI GPT-2 Output Detector from 2019. We found multiple cases where the generated content proved to be original and undetectable (98%). Our research also presents a metric and simple grammatical set for understanding the writing mechanics of chatbots in evaluating their readability and statistical clarity, engagement, delivery, overall quality, and plagiarism risks.

While Turing's original prose scored 14% lower than the machine-generated output, the question of whether an algorithm displays hints of Turing's true initial thoughts remains unanswerable. Our research sheds more light on the importance of establishing a language model in a scientific setting, rather than solely focusing on its ability to deceive human judges. The present work contributes to the larger AI community's efforts in understanding the limitations and potential of large language models in generating text.

XV. Soft-labeling Strategies for Rapid Sub-Typing [16]

We present a new approach to data labeling for object detection in satellite imagery using a semi-supervised learning method with soft labels and label noise. The goal of this research is to automate the process of data collection, curation, labeling, and training for object detection in overhead satellite imagery. Our method takes advantage of the noise in the labels to reduce overfitting and enhance the model's ability to generalize to unseen test data.

Fig. 14: Sub-typing for white vs. non-white vehicles

The experiment involves using a partially trained YOLOv5 model as an initial inference seed to output more refined model predictions in iterative cycles. The model was trained on three unique real-world scenarios, each of which demonstrated how pixel values alone could provide enough information to determine if a car is white or colorful [Figure 14], if a building's roof is blue, or if a crude oil tank is full or empty.

The results showed that the semi-supervised approach using soft labels with label noise improved the model's ability to generalize to unseen test data, rather than just memorizing the training data. In each of the three scenarios, the model was able to accurately identify the object sub-types with minimal human intervention. Overall, our approach provides a cost-effective solution to the challenge of labeling large example datasets for computer vision, and demonstrates the potential of semi-supervised learning with soft labels and label noise for object detection in overhead satellite imagery.

XVI. Soft Labels for Rapid Satellite Object Detection [17]

In this paper, we investigate the use of soft labels in satellite object detection. Soft labels are vector representations of an image's true classification and can be generated by a well-trained model. Our goal was to use soft labels to train an object detection model on satellite imagery and then create a dataset of soft labels.

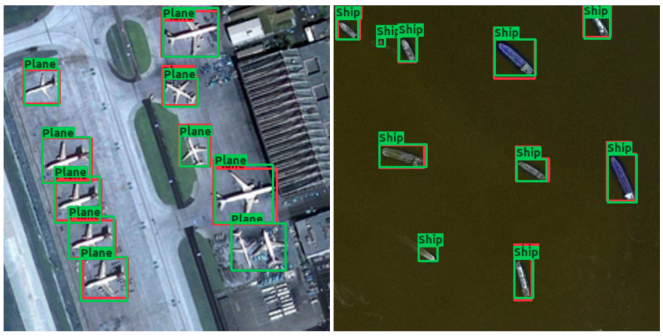

Fig. 15: Ground Truth Labels (green) vs Soft Labels (red)

We proposed using detections as the basis for a new dataset of soft labels. To demonstrate the effectiveness of soft labels, we trained a YOLOv5 model on a subset of the xView dataset to detect cars, planes, and ships. We then used this model to generate soft labels for a second training set, which we trained and compared to the original model [Figure 15]. Our results showed that soft labels can be used to train a model that is almost as accurate as a model trained on the original data.

While soft labels have some limitations, such as missing objects or overfitting, they have several potential benefits. For example, they can automate the process of annotations and save time and resources compared to manual annotation. They can also be used for subtyping, such as breaking down a class into smaller classes. Overall, our results show that soft labels can be a valuable tool for rapid satellite object detection with a less than 6% loss in mAP on our test set.

Conclusion

We have made significant contributions in the fields of computer vision, machine learning, and related areas over the past three years. Moving forward, we plan to continue our research in these areas and explore new frontiers in the field of artificial intelligence. Our goal is to further advance the state of the art and make meaningful contributions that can have a positive impact on society. We are committed to collaborating with other researchers and practitioners to achieve this goal.

Acknowledgment

The authors would like to thank the PeopleTec Technical Fellows program for encouragement and project assistance.

References

[1] . John Schulman, Barret Zoph, "Chatgpt: Optimizing language models for dialogue," 2022. [Online]. Available: https://openai.com/blog/chatgpt/

[2] D. Noever, W. Regian, M. Ciolino, J. Kalin, D. Hambrick, and K. Blankenship, "Discoverability in satellite imagery: A good sentence is worth a thousand pictures," *arXiv preprint arXiv:2001.05839*, 2020.

[3] M. Ciolino, D. Noever, and J. Kalin, "Training set effect on super resolution for automated target recognition," in *Automatic Target Recognition XXX*, vol. 11394. SPIE, 2020, pp. 105–117.

[4] J. Kalin, M. Ciolino, D. Noever, and G. Dozier, "Black box to white box: Discover model characteristics based on strategic probing," in *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*. IEEE, 2020, pp. 60–63.

[5] M. Ciolino, J. Kalin, and D. Noever, "The go transformer: Natural language modeling for game play," in *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*. IEEE, 2020, pp. 23–26.

[6] D. Noever, M. Ciolino, and J. Kalin, "The chess transformer: Mastering play using generative language models," *arXiv preprint arXiv:2008.04057*, 2020.

[7] D. Noever, J. Kalin, M. Ciolino, D. Hambrick, and G. Dozier, "Local translation services for neglected languages," *arXiv preprint arXiv:2101.01628*, 2021.

[8] J. Kalin, D. Noever, M. Ciolino, D. Hambrick, and G. Dozier, "Automating defense against adversarial attacks: discovery of vulnerabilities and application of multi-int imagery to protect deployed models," in *Disruptive Technologies in Information Sciences V*, vol. 11751. SPIE, 2021, pp. 71–78.

[9] M. Ciolino, J. Kalin, and D. Noever, "Fortify machine learning production systems: Detect and classify adversarial attacks," *arXiv preprint arXiv:2102.09695*, 2021.

[10] J. Kalin, D. Noever, and M. Ciolino, "A modified drake equation for assessing adversarial risk to machine learning models," *arXiv preprint arXiv:2103.02718*, 2021.

[11] ——, "Color teams for machine learning development," *arXiv preprint arXiv:2110.10601*, 2021.

[12] M. Ciolino, D. Hambrick, and D. Noever, "Enhancing satellite imagery using deep learning for the sensor to shooter timeline," *arXiv preprint arXiv:2203.00116*, 2022.

[13] M. Ciolino, "Image compression and actionable intelligence with deep neural networks," *arXiv preprint arXiv:2203.13686*, 2022.

[14] M. Ciolino, D. Noever, and J. Kalin, "Back translation survey for improving text augmentation," *arXiv preprint arXiv:2102.09708*, 2021.

[15] D. Noever and M. Ciolino, "The turing deception," *arXiv preprint arXiv:2212.06721*, 2022.

[16] G. Rosario, D. Noever, and M. Ciolino, "Soft-labeling strategies for rapid sub-typing," *arXiv preprint arXiv:2209.12684*, 2022.

[17] M. Ciolino, G. Rosario, and D. Noever, "Soft labels for rapid satellite object detection," *arXiv preprint arXiv:2212.00585*, 2022.