Foundational Vision Models for Mine Detection in UAV Images

Jonas Verbickas
BPTI / Vilnius, Lithuania
jonas.verbickas@bpti.eu

Figure 1: Confidence map overlayed over input RGB image. Brighter white tiles indicate a higher likelihood of explosive being present. Produced using `DINOv2 vit-g reg`.

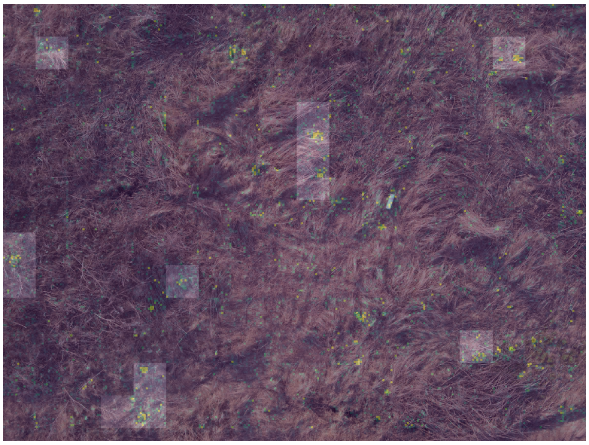

Figure 2: Composite image of the confidence, attention and RGB image. Bright yellow areas are have the highest attention, purple areas have the lowest. Produced using `DINOv2 vit-g reg`.

1 Introduction

Deep neural networks require substantial amounts of data to maintain high performance on novel data (LeCun et al., 2015). For many general computer vision tasks, this isn't a problem due to the availability of large, publicly accessible datasets (e.g., ImageNet, COCO). There are also more specialized public datasets, for example, tracking pedestrians from UAV footage (UAV123, VisDrone). However, when the objective shifts to adapting computer vision neural networks for military applications, the challenge of data scarcity becomes critical because military equipment is highly classified. In such scenarios, options for gathering data on such objects are limited to: 1) ally captured imagery; 2) imagery uploaded on the internet by enemy soldiers showcasing their equipment. This challenge is compounded by the fact that battlefield equipment is continuously evolving and capturing images of such equipment is both dangerous and difficult. In this light, quickly adapting to new equipment with minimal data offers a significant advantage. The task of adapting models to recognize new objects with limited examples is known as few-shot classification. This is optimally addressed using pretrained foundational models which can be tuned to new tasks with just a few examples.

The challenge in acquiring images for various military equipment models also continues with mines. Collecting anti-tank mines is relatively manageable as they are not triggered by human weight, allowing them to be placed on different terrains in secure locations to generate an almost infinite amount of training data. Conversely, finding and safely disarming seismic mines (e.g., POM-2 and POM-3) is nearly impossible. It is reasonable to forecast that in the future more sophisticated mine models will be equally, if not more, challenging to collect safely.

Leveraging drone imagery for mine detection is highly feasible, especially considering the widespread availability of drones in Ukraine. There are several compelling reasons to use them as safe and effective tools for mine detection: 1) Ukrainians already possess a substantial number of drones, eliminating the need for additional investment in

new hardware; 2) drones are operated remotely, which allows for scouting fields without endangering lives; 3) drones can cover and document much larger areas of land quickly compared to on-foot efforts.

2 Related Work

Creating a robust computer vision model that can detect various types of mines from limited data typically requires leveraging insights from previously learned tasks. Foundational models are designed specifically to be adapted for arbitrary tasks with a few examples. These models are trained for tens of thousands of hours on datasets which are too large for humans to label. Foundational models can be subdivided further into 2 subclasses: 1) weakly-supervised models trained on image-text pairs with text captions serving as labels; 2) self-supervised models trained on images only without relying on any labels whatsoever. The former approach was shown by OpenAI with CLIP (Radford et al., 2021) to work very well for knowledge transfer given a large enough training dataset. Whilst the latter approach is shown to work with architectures like MAE (He et al., 2022), DINO (Caron et al., 2021), DINOv2 (Oquab et al., 2024) and I-JEPA (Assran et al., 2023).

CLIP allows for two significantly different approaches of classifying images for downstream tasks. It is possible to use the cosine similarity metric between image and text embeddings in the same way as it was used during the training procedure of CLIP. Additionally, there is an option of "linear probing" i.e. training a linear classifier on top of features extracted by CLIP. According to the findings in the CLIP paper the first approach allows for great zero-shot performance, whilst the second one is known as "linear probing" is able to surpass zero-shot only given enough examples per class. It was found that the number of examples needed per class to surpass zero-shot highly depends on the target classification dataset. We expect our model to require only a few examples per class due to our dataset being similar to EuroSAT and FGVCAircraft datasets. We speculate that just like with EuroSAT, CLIP will struggle to classify our images using cosine similarity due to them being taken from a flying camera with 90-degree angle towards the ground. Furthermore, we expect similar benefits from linear probing as seen in FGVCAircraft, because it consists of highly technical aircraft names (similar to names of mines) unknown to a layman and therefore unlikely to appear in captions of images scraped off of the internet.

In contrast, self-supervised models can only have their knowledge adapted for another task using linear probing. This is due to self-supervised models being trained on tasks that are not useful on their own. This makes only their extracted features the only way of re-purposing their understanding of images.

Both the weakly-supervised CLIP and the self-supervised models mentioned employ (Dosovitskiy et al., 2021) Vision Transformers (ViT) to extract image features, setting the top benchmark performances. The difference lies mainly in their training data and methods. Convolutional networks have also been tested but tend to lag behind ViT in performance.

3 Methodology

3.1 Models

We compare DINOv2 and I-JEPA self-supervised models. We avoid testing MAE altogether because of relatively poor feature transfer to downstream tasks. DINOv2 is chosen over DINO for its advancements in training scale and methodology. Moreover, we test DINOv2 with registers (Darcet et al., 2023) which maintains the performance of DINOv2 while producing attention maps without arbitrary peaks. DINOv2 features a CLS token for linear probing, unlike I-JEPA. To compact I-JEPA's multi-token embeddings into a single token, we average the last layer as it was done by the authors of I-JEPA (Assran et al., 2023).

Considering the rapid developments in deep learning, comparing older weakly-supervised models like CLIP by OpenAI to newer self-supervised models might not yield fair insights. Thus, we opt for the more recent CLIP weights by (Fang et al., 2023), trained on selectively filtered data for enhanced downstream task performance. This model is trained with significantly more data – approximately 5 billion images as opposed to the 400 million used in the original CLIP. The authors demonstrate that their data filtering networks effectively select high-quality data, as evidenced by superior downstream performance using ViT-L image feature extractor when compared to both the original CLIP weights from OpenAI and open-source CLIP variants using larger ViTs (Cherti et al., 2022).

We limit our comparison to the largest avail-

able models, which have proven to perform best in downstream tasks without sacrificing efficiency. For instance, processing a 4000×3000 image, divided into 252 patches of 224×224, takes only 10 seconds on RTX3090 with the largest ViT-giant models.

3.2 Dataset

Our partners have provided (*Visible, Thermal*) aligned image pairs of minefields (4000×3000 for visible and 640×480 for thermal resolutions). These images showcase disarmed mines within a training polygon, predominantly anti-tank for safer collection. Considering foundational models are trained solely on the visible light spectrum, we've excluded the thermal images from our experiments.

The visible light images have a much higher resolution (4000×3000) than the standard (224×224) used in foundational model training. To adapt without losing detail, we slice these into non-overlapping 224×224 patches. This method ensures mines occupy a substantial portion of the image area, making it likely that foundational models will produce embeddings focused on them. Moreover, this tactic allows for parallel processing of an image's patches, maximizing GPU efficiency and speeding up inference.

For every foundational model tested, we precompute image embeddings for our training and validation sets, storing them in the safetensors format for swift access during training. This precomputation streamlines efficient linear probe hyperparameter optimization. The safetensors files categorize patches into one of three classes: *empty, suspicious, explosive* for patches with no annotations, those with potentially misleading objects (like garbage, vehicles, etc.), and those with any explosive (including mines and unexploded ordinances). Patches containing both suspicious objects and explosives are given *explosive* label.

3.3 Visualizing Predictions

To make model predictions interpretable by humans we overlay a grayscale heatmap indicating the "explosive" probability of each patch onto the original image, as illustrated in fig. 1. According to our sources real battlefields contain a lot of debris making the visualization of *suspicious* patches impractical. However, heatmaps inferred using the linear probe are difficult to interpret. Given a patch which is flagged to be "dangerous" it is impossible to understand what pixels lead to that classification. To demystify the model's decisions we visualized the attention maps between *CLS* token and other image patches in the final transformer layer. The attention heatmap generated using this approach effectively creates a low-resolution foreground-background segmentation. Examples of such attention maps from a single attention head can be seen in fig. 3. In easy cases where there is a single obvious foreground object the attention map has an obvious cluster of high attention values at its location. In cases where there are multiple objects in a single patch the attention heatmap is harder to interpret e.g. in the last 2 images where no obvious object was distinguishable the attention is spread more evenly across the image.

However, ViTs have multiple attention heads, with each head evaluating different token channels and producing different attention maps. To visualize everything that the model considers we take the maximum attention between all heads at each token as can be seen in fig. 2.

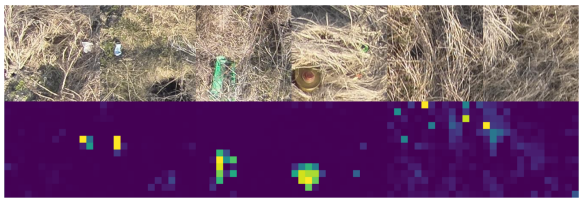

Figure 3: Examples of DINO CLS token attention maps

4 Results

Our initial tests were done using 155k training image patches from 11 training packets (data collected from a single flight) each containing 14k images on average. All large packets with more than 20k image patches were subsampled to 20k in order to force the model to learn various backgrounds. For validation purposes, we selected an image batch that contained the most diverse collection of mines available at the time of testing and subsampled it to 20,000 image patches as well. Additionally, to gauge the model's performance against a substantially different dataset, we created an adversarial Molehill dataset by flying over a Lithuanian countryside field populated with molehills with a different drone. This dataset included anti-tank mine-sized anomalies not seen in the training data. We used the Molehill dataset exclusively as a test set to assess the robustness of models that performed well during validation and not for choosing the top model from hyperparameter optimization.

Model	Val micro-F1	Val macro-F1	Val Binary Accuracy	Molehill Accuracy
I-JEPA vit-h	0.434	0.635	0.826	0.642
I-JEPA vit-g	0.411	0.611	0.819	0.536
DINOv2 vit-l	0.559	0.672	0.850	0.489
DINOv2 vit-l reg	0.473	0.647	0.819	0.355
DINOv2 vit-g	0.546	0.685	0.846	0.525
DINOv2 vit-g reg	**0.606**	**0.697**	**0.853**	0.376
CLIP-DFN-5B	0.524	**0.697**	0.830	**0.733**

Table 1: Metrics from Best-Performing Hyperparameter Configuration of Each Architecture

Hyperparam	Searched Values
Learning Rate	$\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$
Weight Decay	{0.0005, 0.0}
Optimizer	{SGD, Adam}

Table 2: Hyperparameter Search Space.

To measure validation performance we used the F1 score. To combat the fact that $> 95\%$ of patches are of the *empty* class, we compute both micro-averaged and macro-averaged F1 scores. Micro-averaged F1 indicates the model's performance for the validation set i.e. the expected performance if a patch is sampled from the validation set with uniform probability. Whereas macro-averaged F1 score indicates performance on a minefield where all 3 classes of patches are equally likely. Even though such a minefield is unlikely to be seen in the real world it serves as a much better indicator of mine, rather than empty patch, detection performance. To optimize for the macro-averaged performance we subsampled embeddings of each training packet to contain a uniform distribution of classes. The subsampling was repeated for each training epoch to increase diversity in embeddings of the *empty* class.

Hyperparameter optimization focused on validation performance for each architectural model. We conducted a grid-search across values shown in table 2. DINOv2 authors conducted a similar hyperparameter search to measure linear probing capabilities on standard benchmarks. Contrary to their approach, we precompute image embeddings and we could not perform any image augmentations during training.

Results of different hyperparameter configurations that achieved the highest Val macro-F1 for each architecture are presented in table 1. In general, models trained on larger datasets achieved higher Val macro-F1 scores: the lowest score was with I-JEPA models trained on Imagenet-22k with 14M images, whereas DINOv2 trained on 1.2B images and CLIP-DFN-5B on 5B images showed comparable F1 scores. CLIP and DINOv2 vit-g share are tied in Val macro-F1, however CLIP's Molehill Accuracy is an promising sign of its high precision and potentially better generalization. Additionally, we introduce Binary Accuracy metric which measures how often a image patch that has any annotation is assigned

5 Conclusion

Linear probing of foundational vision models achieves solid results without the need for expensive training or custom architecture design. Furthermore, there is a lot of room for improvements. Increasing the training set size and diversity should produce more general models. This can be done both by collecting more data and introducing training image augmentations. Secondly, since the drone flies around in a path where its images have significant overlap, it is likely that even better results could be attained by combining predictions of the same geographic coordinate locations from different viewpoints. Lastly, the expensive fine-tuning of foundational models could be the final step towards squeezing out the most performance from them.

References

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. 2023. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. ArXiv:2301.08243 [cs, eess].

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging Properties in Self-Supervised Vision Transformers. ArXiv:2104.14294 [cs].

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. 2022. Reproducible scaling laws for contrastive language-image learning. ArXiv:2212.07143 [cs].

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2023. Vision Transformers Need Registers. ArXiv:2309.16588 [cs].

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ArXiv:2010.11929 [cs].

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. 2023. Data Filtering Networks. ArXiv:2309.17425 [cs].

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick. 2022. Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, New Orleans, LA, USA. IEEE.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444. Publisher: Nature Publishing Group.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. ArXiv:2304.07193 [cs].

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. ArXiv:2103.00020 [cs].