

Article

Not peer-reviewed version

MuRDE-FPN: Precise UAV Localization Using Enhanced Feature Pyramid Network

[Monika Kisieliūtė](#) and [Ignas Daugėla](#) *

Posted Date: 15 January 2026

doi: 10.20944/preprints202601.1116.v1

Keywords: UAV; satellite; localization; cross-view; transformers; feature pyramid network



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

MuRDE-FPN: Precise UAV Localization Using Enhanced Feature Pyramid Network

Monika Kisieliūtė and Ignas Daugėla *

Vilnius Gediminas Technical University, Antanas Gustaitis' Aviation Institute, Aerospace Data Center

* Correspondence: ignas.daugela@vilniustech.lt

Highlights

What are the main findings?

- We propose MuRDE-FPN, an end-to-end cross-view UAV localization method based on the OS-PCPVT backbone. To enhance the positioning, our decoder includes two novel modules: MultiReceptive Deformable Enhancement (MuRDE) block to enhance the most semantically rich feature layer and Feature Alignment module (FAM), that accounts for spatial misalignment between feature maps.
- The proposed method improved localization accuracy compared to similar methods. The performance increase was noted on both standard and more challenging datasets.

What is the implication of the main finding?

- Our method effectively localizes UAVs in both urban and mixed environments. Moreover, MuRDE-FPN works in both low- and high- altitude settings.
- Given a negligible increase in the computational load, the proposed method can be deployed efficiently in resource-constrained environments.

Abstract

Unmanned aerial vehicles require reliable autonomous positioning beyond the limitations of GNSS, motivating the development of vision-based, end-to-end Finding Point in Map algorithm. This study introduces MuRDE-FPN, an enhanced Feature Pyramid Network (FPN) designed for precise UAV localization, building upon a lightweight one-stream OS-PCPVT transformer backbone. MuRDE-FPN integrates Efficient Channel Attention for adaptive channel recalibration and features two novel components: a MultiReceptive Deformable Enhancement block that utilizes DCNv2 with varying kernel sizes to refine the semantically rich final feature layer, and a Feature Alignment Module for robust layer merging. Evaluated on the UL14 dataset and a new, more diverse UAV-Sat dataset (derived from UAV-VisLoc), MuRDE-FPN consistently outperformed 5 state-of-the-art FPI methods (FPI, WAMF-FPI, OS-FPI, DCD-FPI). It achieved an RDS of 84.26 on UL14 and 63.74 on UAV-Sat, demonstrating superior localization precision. Ablation studies confirmed the cumulative benefits of ECA, MuRDE, and FAM. These findings highlight the effectiveness of custom FPN designs and targeted feature enhancements for UAV-Satellite precise positioning, with MuRDE-FPN providing a robust solution and the UAV-Sat dataset offering a new benchmark for evaluation. Future efforts will address computational efficiency and performance across varying data-quality environments.

Keywords: UAV; satellite; localization; cross-view; transformers; feature pyramid network

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have surged in popularity and in the scope of their applications. While UAVs have been previously used in the military domain, the use of drones in the civilian sector is also booming. Drones are frequently used in agriculture [1], safety and

emergency missions [2], and logistics [3]. Integrating UAVs with reliable autonomous decision-making is crucial for reducing time and resource costs.

To accurately position a UAV, current technology relies on GNSS sensors. This, however, is not reliable in all scenarios: GPS signals are susceptible to natural interference, spoofing, and jamming. To put it simply, there is a need for autonomous positioning technology that is not based on irregular sensor data. While there have been GPS-free approaches proposed to estimate the position of a UAV (such as internal navigation systems, INS, or LiDAR based methods), non-vision algorithms suffer from some key problems. INS is notorious for accumulating drift and requires constant correction from external sensors to maintain positioning accuracy over extended periods [4]. Similarly, LiDAR based mapping approaches, while more accurate, introduce additional constraints due to increased equipment and computational requirements. These issues motivate the exploration of vision-based algorithms either as a substitute or a supplement to such approaches.

With the rise of computer vision (CV) algorithms and availability of remote sensing imagery, vision-based methods have been employed to position a UAV in a satellite. In this context, estimating the drones' position can be achieved using handcrafted feature-matching approaches. These methods are the basis of many vision SLAM pipelines; however, they suffer from poor robustness when viewpoint, illumination, and scale changes. For instance, Scale-Invariant Feature Transform (SIFT) [5], frequently used in vision SLAM to find and match keypoints within images, is notorious for degraded performance under noise, viewpoint and light changes [6]. Some matching-based approaches integrate deep learning (DL) into their pipeline. For instance, [7] proposed using a CNN to extract local descriptors and match them using a graph neural network. Finally, [8] integrated lightweight DL when extracting global features, thus enhancing key-point matching with fused local and global features. However, while these approaches are computationally cheap and fast, they still suffer from poor robustness when viewpoint, illumination, and scale variations are present.

Due to advancements in computer vision, different detector-free localization approaches have been proposed. Currently, there are two main ways to position a drone using such deep learning based pipelines: either retrieve the most similar satellite patch from a satellite image gallery or precisely locate the point within a satellite image (the so-called Finding Point in Map, or FPI, task). While satellite retrieval was the predominant CV-based approach [9–11], it suffers from some drawbacks. First, to achieve localization via the retrieval method on-board, additional preprocessing and disk space is required. Every patch of the satellite image has to have its features extracted, which makes computationally costly as well. Then, similarities for the current view and all of the images in the constructed satellite gallery need to be calculated every time the position needs to be estimated. Moreover, the computational cost of such procedures scale up with the size of the satellite map.

In recent years, an end-to-end precision localization approach was proposed by [12]. The FPI approach aims to solve some of the problems with satellite retrieval method. In particular, it aims to resolve heavy data preprocessing, memory cost, and precise positioning by locating the UAV within the satellite map directly and producing dense heatmap of possible locations. Recently, [13] proposed a one-stream model that drastically reduced models' computational complexity. Since then, research in FPI has shifted from using two-stream methods that are Siamese-like to one-stream backbone extractors and restoring the original output resolution with various decoding strategies. Overall, in FPI, a lot of work has been done in designing better, more lightweight one-stream feature extractors [13–16], as this is where critical advancements in lowering computational cost can be achieved. In this context, the decoding strategy should not be overlooked, as it has been shown that different decoding designs are crucial to overall performance as well. However, the specific decoding strategies usually rely on the particular architecture of the backbone, as some of them do cross-modal modelling in every stage [13,14], and others—only at the lower levels [15,16]. Rather than proposing a new backbone architecture, we suggest adopting most widely used [13,17] OS-PCPVT as a strong baseline. In recent years, the literature on dense prediction tasks has shifted towards enhancing standard FPN-like decoders towards more task-suitable FPN designs. We believe in the potential of

custom FPN refining methods, with special focus on certain scales, particularly for precise UAV localization in challenging, real-world environments.

In any deep learning framework, the scope of generalizability matters as much as the whole method. This makes training and validation data of great importance and as of this day, there is only one dataset that is used throughout all FPI research—UL14 [12]. In other words, only one benchmark dataset exists, as FPI in itself is a relatively new task and also traditional UAV-Satellite pair datasets are insufficient. This hinders the comparison and evaluation of different methodologies, as while the dataset itself is properly constructed, it has its own limitations. The main problem with the UL14 is that the data is very homogenous. This property holds true for both the content (similar architecture and areas) and in quality (UAV flights were conducted in a low-height, satellite imagery shows little to no change). For this reason, evaluating methods on this dataset alone is not sufficient in conducting more rigorous comparison of methods and extending the finding to a more natural setting. While there is some research that extends their evaluation beyond UL14 [14–16,18,19], the comparative datasets are not as rigorous as the UL14, because in those studies only one satellite search map is used [18], the impact of satellite size not on UL14 is not reported [14], or the satellite map is insufficient to extend precision positioning findings towards bigger satellite maps [15,16,18].

In summary, despite recent progress in FPI-based UAV localization, several gaps remain. First, most existing works primarily focus on backbone architecture design, while decoder structures are, while frequently customized, rarely employ a level-aware strategies that explicitly account for the complementary semantic richness of deeper feature layers and the spatial accuracy of the shallower ones. Second, evaluation is overwhelmingly centered on the UL14 dataset, which, while well-constructed, exhibits limited content diversity and relatively small satellite search areas. Third, the behavior of FPI methods under large-area, high-altitude localization scenarios remains under-explored, even though such conditions are common in real-world UAV operations. This work explicitly targets these three gaps. Seeking to enrich scientific literature in UAV localization, our key contributions can be summarized as follows:

1. Seeking to emulate the difficulty and evaluation possibilities of the UL14 dataset, we developed a novel UAV–satellite pair dataset that extends the FPI task. Our dataset, UAV-Sat, retains the quality of different satellite size imagery in the test set, yet the content of the dataset is more general-purpose and representative of diverse UAV applications.
2. Development of a lightweight transformer-based localization framework featuring a novel FPN design that incorporates multireceptive DCNv2 modules to enhance the final backbone feature layer, along with DCNv2-based feature alignment between high- and mid-level feature maps.
3. Extensive comparative analysis of our method on two datasets (UL14, UAV-Sat) and an exhaustive ablation study of our method.

2. Related Work

2.1. Transformer-Based Computer Vision Architectures

The transformer architecture was first introduced [20] and has been widely used ever since. While originally created for natural language processing tasks, transformer-based deep learning architectures have been widely used in computer vision. Generally, any transformer-based deep learning method relies on an attention mechanism that works as follows: given an input sequence, three sets of vectors—queries, keys, and values—are generated through learnable linear projections. For every query vector, an attention weight is computed by calculating the dot product between it and all key vectors and normalizing with softmax function. These weights reflect the relative importance of different input elements. The final output is obtained as a weighted sum of the value vectors, and in this way, the most informative parts of the input are being focused on the most. Unlike convolutional operations, which have a fixed receptive field, the previously described attention mechanism allows for long-range modelling, as queries, keys, and values interact across the entire input sequence, regardless of spatial distance.

With the introduction of the first Vision Transformer (ViT) [21], transformer-based architectures were successfully integrated into the field of vision processing. In ViT, images are divided into patches, then those patches are processed in a transformer encoder using self-attention, just like sequences of text tokens would be processed. While such simple techniques enabled transformer-based architectures to be used in vision processing, ViT was limited in its applicability, as it was tailored for image classification. However, since then, various transformer-based architectures have been developed, each better suited to general-purpose computer vision tasks than the original ViT and, in many cases, competitive or superior to their CNN counterparts.

For instance, [22] proposed a more task-flexible Pyramid Vision Transformer (PVT). Notably, PVT has a hierarchical feature structure, which was made possible by progressively strided patch embedding layers. Given that transformer-based architectures are costly in terms of memory and computational resources, PVT additionally introduced spatial reduction attention, which reduced resource costs while still achieving strong performance. Another hierarchical vision transformer, Swin [23], reduced the computational complexity by replacing global attention with window-based attention. Window-based attention is a variant of attention, where communication between tokens is restricted within fixed-size windows, and this, in turn, lowers computational complexity requirements. Swin specifically does not use global attention and instead captures long-range dependencies gradually across layers by shifting the window partitioning in each consecutive layer. [24] introduced Twins-PCPVT, a PVT with conditional positional encoding (CPE) in every stage after the transformer-encoder layer. In simplified terms, CPE encodes positional information by using depth-wise convolutions to retain spatial relationships between features. Just like Swin, Twins-PCPVT used window-attention; instead of shifting the windows to capture global dependencies, the communication between windows was achieved by using spatially sub-sampled representative tokens. The second version of PVT was introduced [25], where overlapping patch embedding was used, and spatial reduction attention was further improved. Since the introduction of these and subsequent models, transformer-based architectures have been able to perform efficient, multi-scale feature extraction and incorporate transformers into dense prediction tasks.

With transformers, it is also possible to model cross-feature—and by extension cross-image—interactions. A type of attention, referred to as cross-attention, operates with sampling queries from one, and keys/values are from another feature representation. Cross-attention and its variants have been used in cases, where cross-modality interaction is inherent to the task, such as image-to-image matching or multi-view perception. For instance, both Matchformer [26] and LoFTR [27] employ self- and cross- attention to explicitly match features within and between image pairs and thus enabling robust correspondence estimation. Likewise, OSTRack [28] applies the same principle of using both attention mechanisms to create a one-stream framework between template and search images for object tracking. These examples demonstrate that transformer-based algorithms are effective in extending their usability not only for feature extraction but also for modelling early and task-specific feature interactions across multiple inputs.

2.2. Feature Pyramid Networks

Feature Pyramid Networks as a specific design choice have been used in a variety of cases. As a modelling structure, FPNs are used when a task needs strong performance across scales or when task requires pixel-level performance (as in segmentation or depth estimation). Feature Pyramid Networks [29] were proposed to mitigate natural loss of spatial details when using modern CNN-based (and more recently, transformer-based) vision models. Typical FPN exploits multi-scale feature representations and constructs such feature representation that preserves semantic richness, spatial resolution, and is suitable for subsequent downstream tasks.

The core idea behind any FPN is as follows: instead of using the last and most spatially compressed feature layer, integration of previous feature maps should not be overlooked. For instance, in the seminal FPN [29] work, it is suggested to construct FPN in such a way: given the hierarchical nature of any backbone network, feature maps from different levels are combined with

a top-down pathway using lateral connections. In them, higher-level features are progressively upsampled and merged (in [29]—a simple addition operation is sufficient for a merge) with lower-level (and higher-resolution) ones. In recent years, FPNs have evolved significantly, with main differences in architectural design being those of the pyramid topology and micro architectural design choices (such as using different feature upsampling, refinement, and merging logic).

For instance, [30] introduced PANet—a FPN with an additional bottom-up path, adaptive feature pooling, and fully connected fusion. Later, NAS-FPN [31] introduced dynamic FPN topology search—instead of using a predetermined architecture design, NAS-FPN used a controller network to generate FPN architecture dynamically. In essence, to get the optimal design, a recurrent neural network was trained—its input was sampled from candidate FPN architectures (for instance, feature level selection, merging operation) and its reward was defined as selected architecture's performance on the validation set. While such a method resulted in an improved performance compared to hand-crafted FPNs at the time, it also incurred higher computational complexity. Later, [32] introduced BiFPN—a bidirectional FPN that included various design choices. Just like the PANet, it includes a bottom-up path, but like NAS-FPN, the blocks of top down are repeated. Nodes with a single input are removed, and features are fused with learnable fusion.

FPN architectures can sometimes be designed asymmetrically, as in certain cases, there is prior knowledge that selectively enhancing specific feature levels can be more beneficial than uniform processing. For instance, to improve semantic segmentation of maritime images, [33] used an information interaction module, that explicitly integrates raw images with high-level feature representations (in this case—the last two layers). These three inputs are jointly processed, and a new feature map is generated, which is integrated into the top-down FPN pathway. [34] also used a highly asymmetric decoder (top-down FPN path) structure with multiple architectural enhancements to better image segmentation. First, instead of the usual channel unification with typical convolution with kernel size 1, [34] used squeeze and excitation blocks to adaptively reweight channels. Second, global semantic branch processes inputs from the last two feature extractor layers.

In certain circumstances, it is not unusual to find improvements not only in FPN topology, but also in the introduction of task-specific FPN enhancements. For example, in medical image segmentation, retaining or extracting boundary information is key to improving performance. This reasoning is why [35,36] both used reverse attention mechanism—a technique that controls how information flows and can help enhance specifically edge information—in their decoder (top-down) paths. Another example of extending pyramid structures in enriching feature representations for the task at hand is provided by [37]. In their work, authors tackle small object detection and, to enhance feature richness, [37] employ multireceptive-field-based enhancement modules. Interestingly, these modules are used only on specific feature maps (the first two least spatially reduced ones), which again showcases an asymmetric FPN modelling. Such findings in a broader context imply that task-specific FPN design choices, especially those that extend FPN beyond topology, are of great importance to overall model effectiveness.

2.3. Finding Point in Map

To enable more precise UAV localization, [12] proposed an end-to-end dense prediction approach. First, the authors outlined two relevant metrics—Relative Distance Score (RDS) and meter-level accuracy—and formulated UAV localization as a dense prediction task, called Finding Point in Map. In FPI, the final output of a model should be a satellite-sized grid of probabilities that note whether a point on the map is a central UAV location. To achieve such an output, two backbones of DeiT-S [38] (backbone for UAV and satellite) are used as feature extractors, then the interaction between these features is calculated using Siamese cross-correlation. Authors additionally implement a true positive generation technique to balance low true positive cases—instead of treating only one point as the only possible true positive, they sample an area around ground truth (creating a synthetic true positive grid). Then, weighted balance loss is used to additionally counteract the imbalanced nature of this task.

Following the FPI approach, [39] proposed WAMF-FPI, a two-stream network with PCPVT-S [24] as a backbone feature extractor, feature-pyramid neck, and weighted feature fusion. The WAMF-FPI method differs from the original FPI method in that it uses a hierarchical vision transformer to extract features at different levels, as opposed to just using the last layer, where features are semantically rich but also lose their positional information. Additionally, cross-correlations between UAV features and last satellite feature maps are calculated, and weighted fusion is used. According to [39], these techniques enable to prevent spatial information loss, that was prevalent in the original FPI approach. Moreover, authors propose a new loss weighing method—Hanning loss—where true positives are weighed according to their distance to the central point. This allows models to focus more on the central area rather than treating all true positive points the same. WAMF-FPI outperformed the original FPI approach on UL14 dataset.

[13] proposed a single-stream transformer-based feature extractor, OS-PCPVT, that leveraged both self and cross attention to capture early feature interaction. Using such a backbone design greatly reduced model size, which is especially relevant with regard to on-board deployment. Authors additionally used atrous convolution after feature extraction to capture relevant pixel-level information and improve restoration. Most notably, [13] employed multi-task training to predict not only the final heatmap, but also an additional offset for vertical and horizontal direction. For this reason, a combined loss function was used, where classification was evaluated using Hanning loss, and smooth L2 loss was used for regression task. Similarly, [17] too leveraged the OS-PCPVT backbone, with emphasis on fusing features across different stages. Instead of using standard FPN, [17] included ASFF-like feature fusion, where spatial weights were generated for each feature map. Additionally, a custom head was used to achieve optimal spatial restoration; it included a key module composed of deformable convolutions that progressively reduced the channel count and increased the spatial dimension.

Latest research in FPI involves [15], who proposed their own single-stream pyramid transformer (SSPT) model that, like OS-PCPVT, is a one-stream and hierarchical. Notably, SSPT does not employ cross-attention at every stage, it is applied only at the final one. Crucially, the allocation of self-attention in the first two backbone stages and cross-attention in the last stage yields the best results. Then, the final heatmap is also restored from only that same last feature layer. To avoid positional loss, progressive upsampling and a multi-scale pyramid scheme were used. Notably, authors also proposed Gauss loss, which is similar to Hanning loss except that the weighting is performed a Gaussian window function. Expanding upon SSPT, [16] included a channel-reduction attention mechanism in the backbone and designed a residual feature pyramid network (RFPN), and [14] combined both self- and cross-attention at every feature extraction stage with a symmetrical feature pyramid network, allowing for adequate incorporation of the most semantically rich feature layer. A slightly different approach was proposed by [18]: instead of using a traditional vision transformer, [18] incorporated vision Mamba to capture long range dependencies. Additionally, authors used consistency regularization training with center masking augmentation so that the model would learn to predict the output even when central UAV features are missing.

3. Materials, Method and Evaluation

3.1. Datasets and Preprocessing

To evaluate our method, we utilized two publicly available UAV-Satellite pair datasets: UL14 [12] and UAV-VisLoc [40]. UL14 dataset was left as is, however, from UAV-VisLoc, we created our own, UAV-Sat dataset. While evaluation on UL14 is crucial for evaluation as a benchmark dataset for the FPI task, the addition of UAV-Sat allows to extend our findings to a setting in which satellite maps are comparatively larger and more variable and the UAV images are sampled from a higher altitude. In the following subsections, we outline the specifics of each dataset and our data preprocessing configuration. Both the UL14 and UAV-Sat datasets were created so that the validation and test sets include augmented images. This makes the method's validation much more reliable.

3.1.1. UL14 Dataset

UL14 consists of UAV-satellite image pairs from 14 different university campuses in China, 10 of which are for training and 4 for testing. The training set comprises UAV images at 512x512 pixels, and satellite images at 1280x1280 pixels. The test set contains UAV images at 256x256 pixels, and satellite images at 768x768 pixels. The satellite imagery in this dataset has a spatial resolution of 0.294m/pix. Unprocessed training set satellite imagery is centrally aligned, meaning that the UAV is at the center of the paired satellite patch. Satellite images from the test set, however, do not share this quality—they are not centrally aligned and also have varying sampling scales. In test set, for each UAV image, there are 12 satellite patch images, in all of which UAV location is randomly distributed, and satellite scales vary from 700 to 1800 pixels.

UAV images are sampled from 80-, 90-, or 100-meter heights, which makes this dataset low altitude in terms of UAV data. Most satellite images in UL14 exhibit a relatively straight-down view, with little visible tilt. Qualitatively, since UAV images are sampled from university campuses, this dataset is urban-focused, with no suburban areas or challenging nature environments (such as deserts, valleys, mountains). Together with the absence of any noticeably off-nadir satellite imagery, UL14 is a comparably limited dataset for general-purpose UAV positioning applications.

3.1.2. The Creation of UAV-Sat Dataset

To evaluate and enhance the applicability of our method and to address the limitations of UL14 dataset, from the publicly available UAV-VisLoc dataset we created our own, UAV-Sat dataset that aligns with the FPI task. Originally, UAV-VisLoc is comprised of UAV-satellite image pairs and is collected from 9 various regions in China. Differing from UL14, this dataset is more diverse in terms of visual content: UAV-VisLoc contains images from both man-made (urban and suburban) and natural (river valleys, farmlands, mountainous areas) environments. UAV flight heights range from 405 to 840 meters, making this dataset more applicable in evaluating the performance in high-altitude scenarios. Compared with satellite imagery in UL14, UAV-VisLoc satellite images exhibit a greater amount of change, because of the time difference between satellite and UAV capture. This makes UAV-VisLoc better suited for evaluating methods under realistic conditions in which satellite imagery may be suboptimal. The satellite imagery in this dataset has a spatial resolution of 0.3m/pix. From UAV-VisLoc we created our own UAV-Sat dataset by following such procedures:

4. Inclusion of UAV-satellite pairs within the whole dataset: we chose to exclude pairs that contained UAV images with uninformative or non-pertinent features. For this purpose, UAV images were centered-cropped, and if the images were deemed uninformative after this procedure, they were excluded from the final dataset. We consider an image to be informative if it contains some sort of permanent feature (e.g., a docking station on a shore, some road within the forest, permanent natural landmarks). We also excluded pairs with satellite images for which it was not possible to crop them to a 3500x3500 pixel range without padding.
5. Train, validation, and test set splitting: after initial image selection, training, validation, and test sets were constructed, splitting the whole dataset into training and test sets (85/15 ratio) and then splitting the training set again into training and validation sets (90/10 ratio). Final splitting proportions were 76.5/8.5/15.
6. Train set preparation: for the training set, we chose to crop satellite images to 3500x3500 pixels to get enough coverage area for Random Scale Crop (RSC, see 3.1.2) augmentation and enrich the final dataset with bigger satellite coverage images. Such image cropping yields image pairs that cover around 1 square kilometer satellite area. UAV images were not processed further (the same center crop from step 1 was retained).
7. Validation and test set preparation: as in the training set, we also cropped satellite images to 3500x3500 pixels and retained center crop of UAV from step 1. Additionally, each satellite image underwent RSC augmentation in a similar manner to the UL14 dataset. For every image, we constructed 12 satellite images of varying sizes, with minimum satellite image size being 2400

pixels and maximum being 3500 pixels. Notably, each of those 12 images had its target location pixel randomly sampled.

8. Image format and saving: after initial processing (steps 1-4) images were saved to a fixed resolution of 512x512 and 1280x1280 for UAV and satellite images, respectively, in the training set and 256x256 and 768x768 for UAV and satellite images, respectively, in the validation/training sets.

These steps were taken to ensure that our UAV-Sat dataset would be as comparable in evaluation difficulty to the UL14 as possible. Table 1 and Figure 1 highlight a comparison between the two datasets.

Table 1. Comparison between UL14 and UAV-Sat datasets.

	Dataset					
	UL14			UAV-Sat (ours)		
	Sat	UAV	Satellite cover area*, km ²	Sat	UAV	Satellite cover area*, km ²
Train	6768	6768	0.1475	3330	3330	1.1025
Test	27972	2331	0.0441-0.2916	7836	653	0.5184–1.1025

* The satellite area for the test set is a range of values, since in both datasets there are 12 satellite images of different scales.

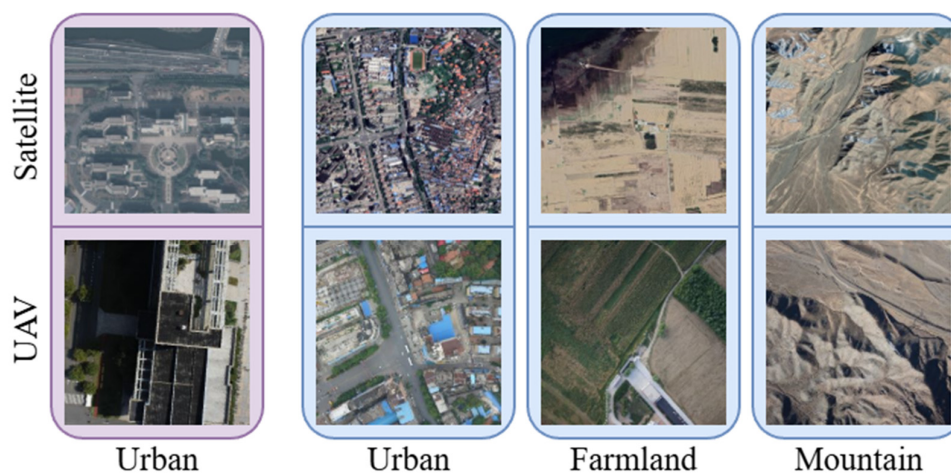


Figure 1. UAV and satellite views from the UL14 and UAV-Sat datasets. Purple outline indicates pair from UL14 dataset and blue – from UAV-Sat dataset.

3.1.2. Data Preprocessing

As previous studies have done, in the data preprocessing module, we included Random Scale Crop [12] (visualized in Figure 2). Given a centered image (i.e., an image where the target is at the center), this augmentation selects a random new point and crops a specific area around it. The coordinates of the true location are then recalculated, and the target location is no longer centered. Instead, it appears at a random location within the cropped area. This augmentation has two main parameters: `map_size` and `cover_rate`. `Cover_rate` parameter controls the range around the centered true location where a new random new point can be sampled, and the `map_size` controls how much of the area around randomly sampled dot is covered.

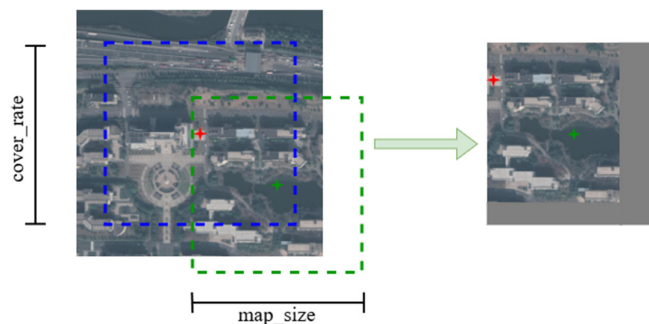


Figure 2. Random Scale Crop visualization. Red star denotes true location of UAV, green—center coordinated of an augmented image.

In our case, this specific augmentation was used to expand both the training sets (RSC applied randomly with respect to cover area) and the validation/test sets (the same augmentation applied statically, with cover area fixed to a set of predefined sizes). By applying RSC in such a way, we ensure that there is no center bias during training and that our validation/test sets are more complex to classify, and thus our evaluation is more reliable.

Additionally, our data preprocessing pipeline includes an image normalization module, as we scale our images to be in the range of $[-1, 1]$. This type of normalization was selected since our initial experiments (conducted only on UL14) yielded superior performance compared to min-max scaling (value range $[0, 1]$). Such preprocessing was performed during the training, validation, and testing phases.

3.3. Backbone Network

To conduct our experiments, we leveraged pretrained OS-PCPVT [9] backbone as our feature extractor. OS-PCPVT is a transformer-based hierarchical network (outlined in Figure 3), that utilizes both self-attention and cross-attention to extract relevant features and their interactions early. Moreover, it's a dual-input network that processes both UAV and satellite images simultaneously.

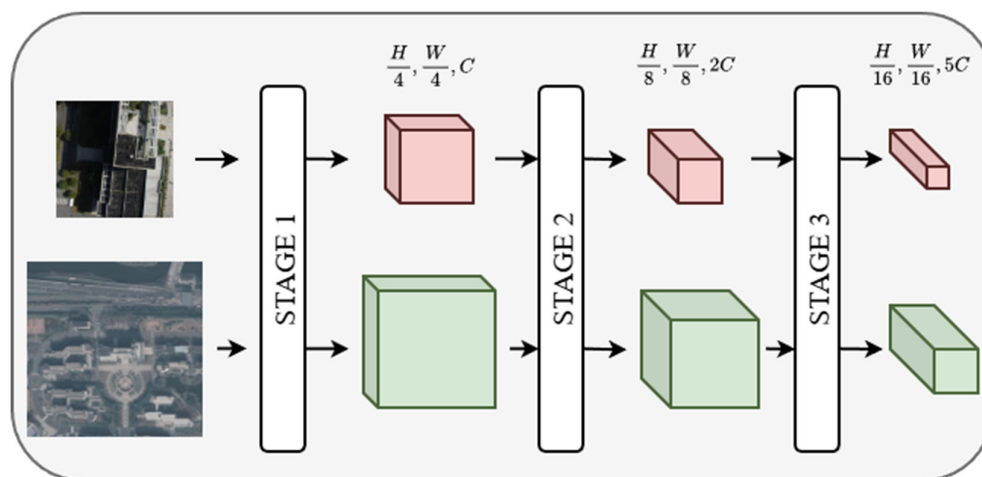


Figure 3. General outline of the OS-PCPVT [13] backbone. Red layers indicate UAV branch, green ones—satellite branch.

Every stage of OS-PCPVT works as follows: both inputs undergo patch embedding operation after which all tokens are concatenated and enter the transformer block. After the first transformer block, the positional encoding generation is performed and then the transformer block is applied N times. N values are 2, 4, and 6 for stages 1, 2, and 3 respectively. Just like PVT, OS-PCPVT has spatial

reduction attention, which keeps the backbone light. In the transformer block, self-attention is performed for the UAV branch (keys, values and queries are sampled only from UAV input), and cross-attention is performed for the attention branch (keys and values are sampled from UAV and queries are sampled from satellite input). In such a way, the UAV branch exclusively models the UAV features, and satellite branch—the interaction between UAV and satellite.

3.4. MuRDE-FPN

To keep our model lightweight, we limit ourselves to modeling only on the satellite branch from the OS-PCPVT backbone. We are able to do such modelling, given that the output from every stage in the satellite branch is constructed using cross-attention. In traditional FPN, the fusion of different scales is achieved as follows: each layer is unified with a convolutional layer with kernels of size 1, the current and smaller (in terms of spatial dimensions) feature maps are upsampled and then added (or concatenated) with the previous layer. We aimed to enhance the classical FPN structure so it would better serve the FPI task. The central design principle of MuRDE-FPN is selective refinement of the most semantically rich but spatially degraded feature level, rather than uniform enhancement across all pyramid levels. Additionally, traditional fusion methods (addition, concatenation, even weighted fusion) are insufficient to model cross-scale interactions, and for this reason MuRDE-FPN also includes a spatial alignment module. The general outline of our architecture is visualized in Figure 4.

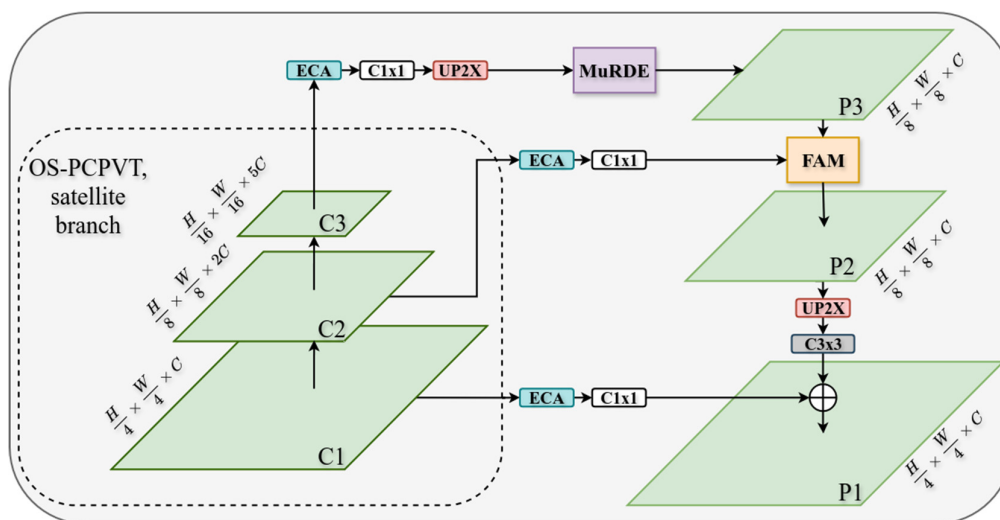


Figure 4. Our enhanced FPN, MuRDE. C1x1 and C3x3 denote convolutional layers with kernel sizes 1 and 3, respectively; UP2X marks a feature upsampling operation using bilinear upsampling. ECA marks efficient channel attention.

First, before lateral channel unification, we included efficient channel attention (ECA) [41] operation for every feature map. ECA is a lightweight channel reweighting technique that adaptively suppresses or emphasizes channels without dimensionality reduction. In practice, ECA performs average pooling to extract channel-wise spatial descriptors, which then undergo one-dimensional convolution to capture local cross-channel interactions. We used channel-adaptive kernel selection, as proposed in the original implementation. Final channel-attention weights are normalized using sigmoid function. Including ECA before channel unification allows for a more effective channel recalibration. We specifically chose this channel reweighting method in order to keep our final computational complexity almost intact, as ECA is very lightweight.

Then, in our design, we included two key blocks: MuRDE (MultiReceptive Deformable Enhancement) and FAM (Feature Alignment Module). In both modules, Deformable Convolution [42] is a key operator to achieve our goals. This type of convolution is an extension of standard

convolution with the inclusion of learnable spatial offsets. Output of such a convolution y at a location p is calculated:

$$y(p) = \sum_{k=1}^K w(p) \cdot x(p + p_k + \Delta p_k) \quad (1)$$

where p_k is sampled from a regular convolution sampling grid at a location k , and Δp_k denotes learned offset for the same sampling point k . In other words, this type convolution is capable of adapting to input and capturing irregular patterns. In its second variant, DCNv2 [43], a modulating mask Δm_k is introduced:

$$y(p) = \sum_{k=1}^K w(p) \cdot x(p + p_k + \Delta p_k) \cdot \Delta m_k \quad (2)$$

Modulation masks are also learnable, which means that DCNv2 is additionally capable of controlling the contribution of each sampling location. We outline the comparison between the computation of DCNv1 and DCNv2 in Figure 5.

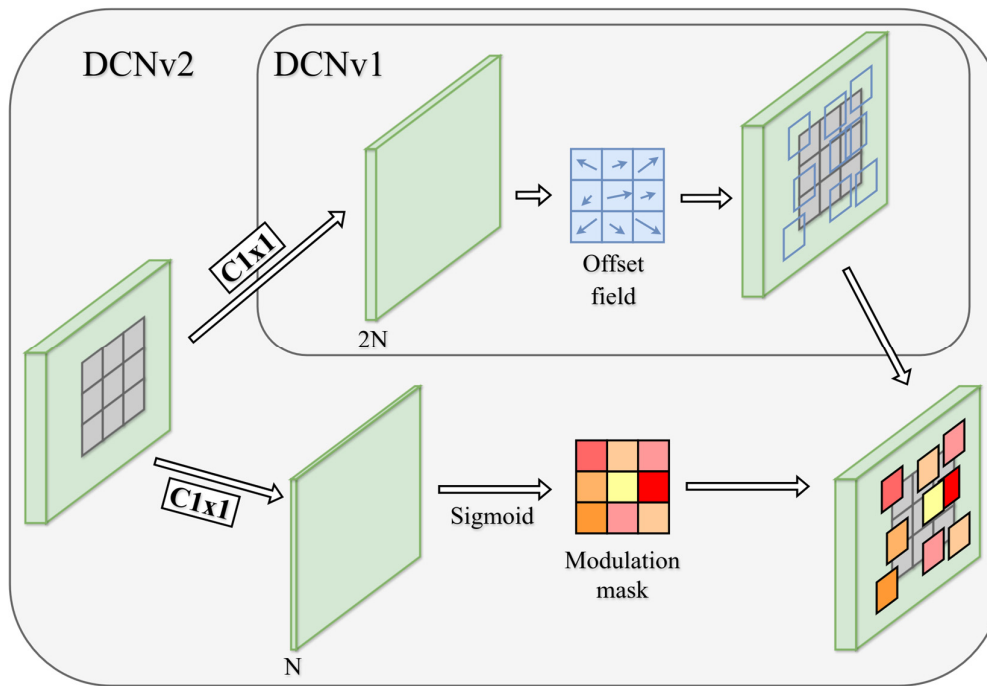


Figure 5. DCNv1 and DCNv2 operations. N and $2N$ denote the number of channels, where N is kernel size squared.

To enhance and refine the last layer of the backbone, we specifically designed MuRDE block. While the last layer from the satellite branch in OS-PCPVT is most semantically rich, it suffers from spatial loss, since the original satellite input is downsampled 16 times. Because of this, we believe that additional refinement and enhancement of such a feature map are beneficial. To achieve this, MuRDE block uses DCNv2 as follows: given an input feature map, different configurations of DCNv2 were computed. Configurations differ in kernel size and dilation, or, in other words, MuRDE block is able to capture multiple resolutions. Outputs from every DCNv2 convolution were then fused by simple addition, followed by Group Normalization and ReLU activation function (shown in Figure 6). Final configuration of parameters in MuRDE block can be seen in Figure 6, and our final configuration of MuRDE module includes convolutions, that have receptive fields of 3×3 , 5×5 , and 9×9 .

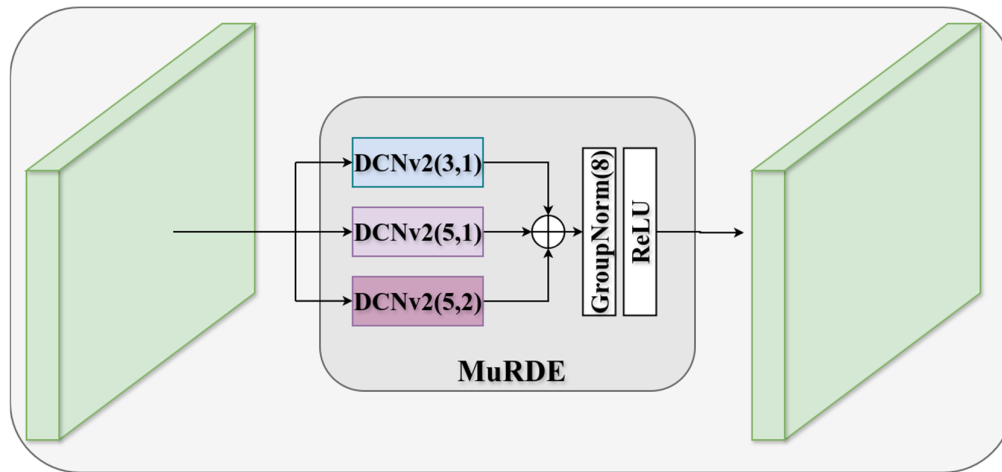


Figure 6. MuRDE module. On the right side, the final MuRDE configuration is shown.

Inspired by [44–47], our final model also has a Feature Alignment Module (FAM) (Figure 7). We include FAM only to reduce the misalignment of the last two feature layers (or specifically, C2 output from OS-PCPVT and output from MuRDE module), as in our case, including this module between the first two (C1 and C2) actually degraded the performance.

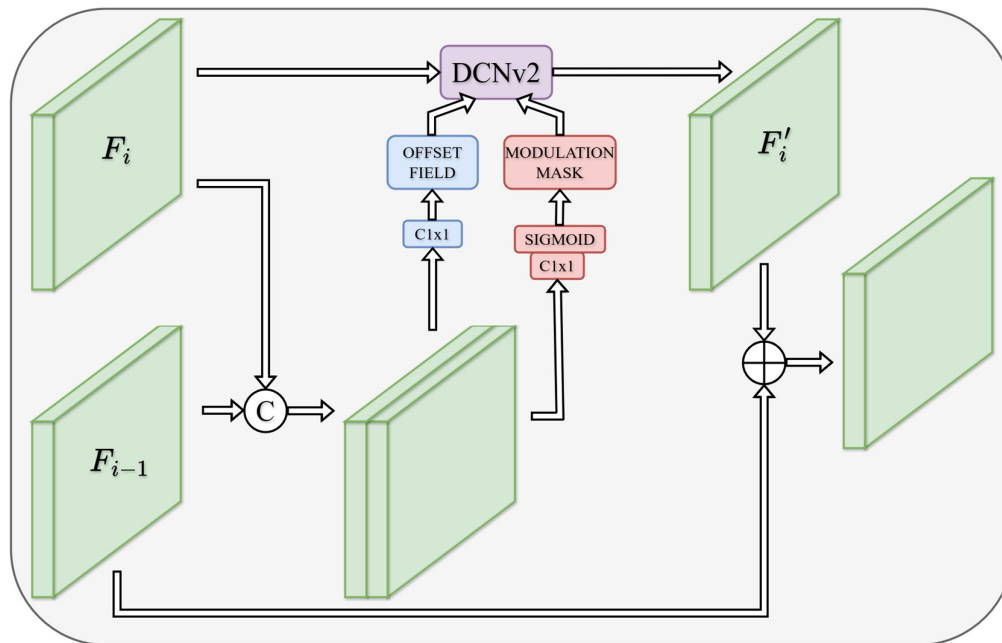


Figure 7. FAM module.

The FAM module works as follows: given two feature maps, those coming from deeper in the backbone are prone to more misalignment. To address this, the two feature maps are concatenated, and offsets and modulation masks are computed for the deformable convolution. Then, the map that suffers from more misalignment (in our case—output from MuRDE block, as it was derived from a more downsampled feature layer) is refined with deformable convolution. Such now aligned map is fused with another map, in our case, simple addition was used.

3.5. Experimental Setup and Evaluation

Our experiments were conducted on an NVIDIA GeForce RTX 4090 GPU. Since a pretrained OS-PCPVT backbone was used throughout all of our experiments, we set discriminative learning rates: 5e-5 for the backbone and 1e-4 for the head and neck with the AdamW optimizer, with its parameters set to default values. To achieve optimal LR scheduling, our method uses MultiStepLRScheduler with milestones set to 10, 14, and 16 and gamma parameter set to 0.2. We trained our model for 30 epochs with a batch size of 8 using a Hanning window loss [39] with R set to 33. To evaluate other FPI methods on the UAV-Sat dataset, we opted not to use the same training hyperparameters as for our model but instead we followed procedures that these models use in their own setting. This was done in order to give each method the best shot possible, as in training DL algorithms, the choice of various hyperparameters matters a lot, and we believe that each model was optimized best by their authors.

In line with the current FPI research, we evaluated our and other methods using relative distance score (RDS) and meter-level accuracy (MA@K) metrics. RDS can be defined as:

$$RDS = e^{-k \times RD} \quad (3)$$

where k is a scaling factor (set to 10), and RD is relative distance between the predicted image location (X_p, Y_p) and ground truth image coordinates (X_g, Y_g) :

$$RD = \sqrt{\frac{\left(\frac{dx}{w}\right)^2 + \left(\frac{dy}{h}\right)^2}{2}} \quad (4)$$

Above, $dx = |X_p - X_g|$, $dy = |Y_p - Y_g|$, and w and h are width and height of the image. In this way, RDS is a crucial metric for evaluating models' performance at the pixel level and is resistant to image scale transformation. Note that higher values of RDS are better, and the score is distributed between 0 and 1. This is the main model optimization metric as well. However, RDS is not very helpful when evaluating real-life distance performance. For this reason, an additional metric, MA@K is defined as:

$$MA@K = \frac{\sum_{i=1}^N 1_{SD < Km}}{N} \quad (5)$$

Here $1_{SD < Km}$ is a condition defined as:

$$1_{SD < Km} = \begin{cases} 1, & \text{if } SD < Km \\ 0, & \text{if } SD \geq Km \end{cases} \quad (6)$$

where SD denotes and Km marks chosen meter threshold. Note that MA@k is sensitive to scale. For larger images, resizing operation gives way to larger unit pixel values in meters. This in turn means that small distances between predicted and true pixel locations might be small, but meter distances—larger compared to smaller images.

5. Experimental Results

In this section, we present our findings when comparing our approach with 4 other FPI methods—FPI, WAMF-FPI, OS-FPI and DCD-FPI. Two of these (FPI and WAMF-FPI) have an unshared backbone (feature extraction is done independently and feature interaction modelling comes after) and another (OS-FPI and DCD-FPI) have a shared backbone. Furthermore, OS-FPI and DCD-FPI have the same backbone as ours (OS-PCPVT), which makes such evaluation valuable in enabling comparison of non-generic FPN-based methods. All methods are first compared on the UL14 dataset to ensure comparability with prior work, and subsequently trained and tested on the UAV-Sat to assess robustness beyond the standard setting.

5.1. Evaluation on UL14 Dataset

Table 2 outlines the comparative performance on the UL14 dataset, and it is seen that our method, MuRDE, outperforms them all. Specifically, we outperform comparative models (OS-FPI

and DCD-FPI) by 8.01 and 7.11 in RDS metric. Looking at the aggregated metrics, our method is also superior in meter accuracy at 3, 5, and 20 meters (5.84, 8.39 and 9.57 respectively) compared to DCD-FPI. While our method outperforms both of the two-stream methods and OS-FPI in terms of parameter count and GFLOPs, DCD-FPI is still comparatively lighter. Specifically, our method is heavier by 0.18M parameters and has 0.27 more GFLOPs than DCD-FPI. Nonetheless, we believe that such an increase in computational complexity is tolerable given the overall performance of MuRDE-FPN on the UL14 dataset.

Table 2. Comparison of different FPI methods, evaluated on UL14 dataset. We report MA@k as a percentage. Bold indicates the best and underline – the second best performance.

Method	RDS	MA@3	MA@5	MA@20	Params (M)	GFLOPs
FPI [12]	57.22	-	18.63	57.67	44.48	14.88
WAMF-FPI [39]	65.33	12.49	26.99	69.73	48.5	13.32
OS-FPI [13]	76.25	22.81	44.31	82.52	14.76	14.28
DCD-FPI [17]	<u>77.15</u>	<u>25.09</u>	<u>47.03</u>	<u>83.39</u>	13.96	11.54
MuRDE-FPI	84.26	30.93	55.42	93.06	<u>14.15</u>	<u>11.81</u>

To provide a more extensive evaluation on UL14 and the impact of satellite size towards the precision metrics, Figure 8 illustrates MA@k dependence on satellite image size to the performance of those methods, that utilize OS-PCPVT backbone. Interestingly, our model shows inferior performance than DCD-FPI of MA@3 metric for satellite maps, that are smaller than 1500x1500 resolution. However, MuRDE FPN outperforms both OS-FPI and DCD-FPI in all other cases and in all other metrics.

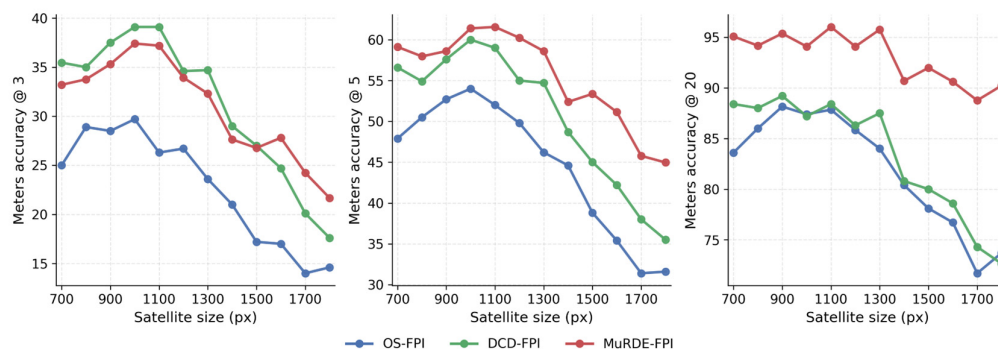


Figure 8. Impact on performance depending on satellite template size, UAV-Sat dataset. We compare methods with the same backbone (OS-PCPVT): OS-FPI [13], DCD-FPI [17], and ours, MuRDE.

5.2. Evaluation on UAV-Sat Dataset

To evaluate the performance of our method in a more general setting environment, we also provide a comparison on our UAV-Sat dataset (Table 3). Given the fact that UAV-Sat dataset consists of high-altitude coverage and bigger map sizes, we report RDS as the main metric, but also additionally provide MA@k metric for higher ranges than for UL14 dataset. In particular, given that the edge of the biggest evaluated satellite map is approximately 3 times bigger in UAV-Sat than in the UL14, we report MA@k starting at 10 meters rather than 3 meters. In Table 3, it can be seen that all methods suffered a substantial decrease in performance when comparing RDS scores to those of the UL14 dataset: -19.61 for FPI, -21.38 for WAMF-FPI, -17.22 for OS-FPI, -20.71 for DCD-FPI, and -20.52 for MuRDE-FPN. The average decrease in performance is 19.89 RDS points, which attests to UAV-Sat difficulty. Even when evaluating different methods on such a dataset, our method outperforms comparative methods in the RDS metric: we see a 4.71 point increase when compared to OS-FPI and a 7.3 point increase when compared to DCD-FPI. Interestingly, the performance gap

between OS-FPI and MuRDE is significantly lower when evaluated on the UAV-Sat dataset than on the UL14 dataset. An increase in MA@k scores can be seen as well.

Table 3. Comparison of different FPI methods, evaluated on UAV-Sat dataset. We report MA@k as a percentage. Bold indicated the best, and underline—the second best performance.

Method	RDS	MA@10	MA@20	MA@30	MA@40	MA@50
FPI [12]	37.61	0.15	0.64	1.2	1.8	2.6
WAMF-FPI [39]	43.95	0.18	0.69	1.4	2.3	3.5
OS-FPI [13]	<u>59.03</u>	0.26	1.1	<u>2.1</u>	<u>3.3</u>	<u>4.5</u>
DCD-FPI [17]	56.44	<u>0.33</u>	<u>0.94</u>	1.8	3.1	4.1
MuRDE-FPI	63.74	0.38	1.1	2.4	3.8	5.3

In Figure 9, the impact on RDS depending on satellite image size can be seen. In this regard, our method is also superior to comparative methods (OD-FPI and DCD-FPI).

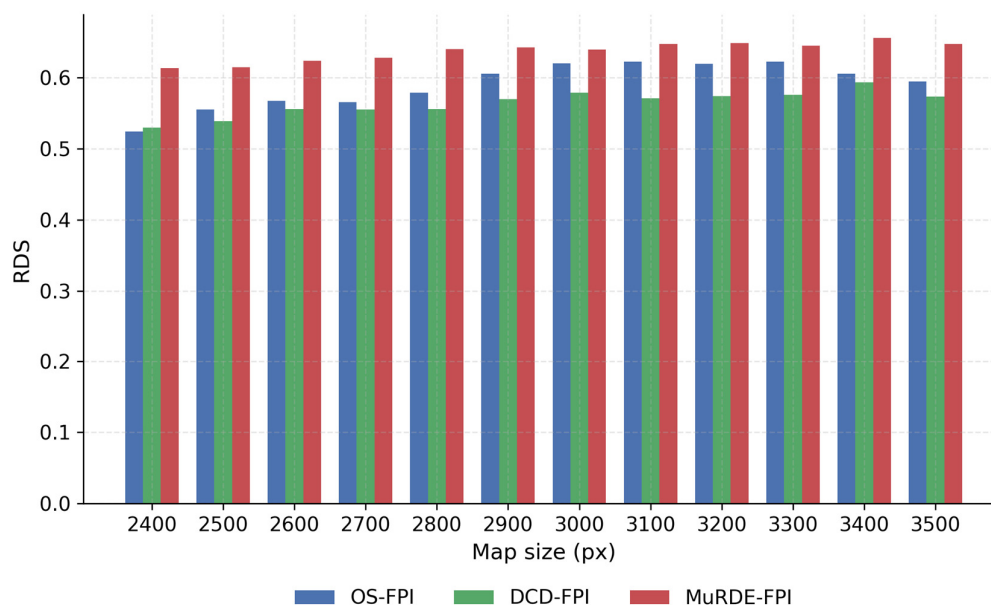


Figure 9. Impact on performance depending on satellite template size, RDS evaluated on UAV-Sat dataset. We compare methods with the same backbone (OS-PCPVT): OS-FPI [13], DCD-FPI [17], and ours, MuRDE.

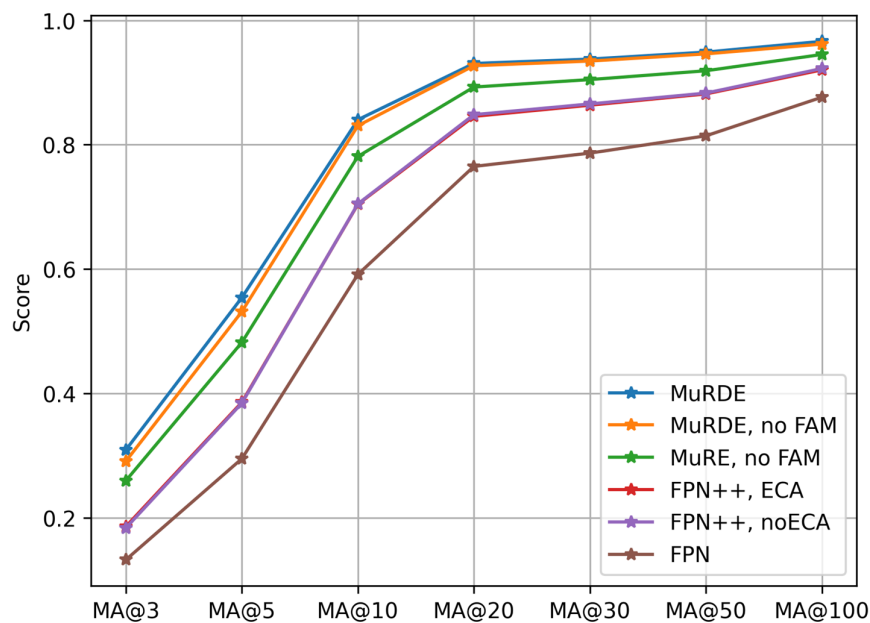
6. Ablation Study

To further investigate the impact of different fusion, refinement, and enhancement strategies on performance, we conducted an ablation analysis on the UL14 dataset. In the first analysis, we compare our strategies in an additive manner—we selected our baseline to be a simple feature pyramid network (we denote it FPN) as described in [29], and with FPN++ we denote a variant of traditional FPN, where an additional refinement convolution of kernel size 3 was added after upsampling. Next, we additively introduce our proposed steps: efficient channel attention (ECA), MultiReceptive Enhancement (MuRE), MultiReceptive Deformable Enhancement (MuRDE) and the addition of feature alignment (FAM), but only to fuse output output of MuRE/MuRDE and C2. In MuRE, the enhancing logic is the same as our method, except we use classic convolutions here. General results on UL14 are shown in Table 4.

Table 4. Comparison of methods, evaluated on UL14 dataset.

Method						RDS	GFLOPs	Params (M)
FPN	FPN++	ECA	MuRE	MuRDE	FAM			
+						68.94	10.4	13.51
	+					76.12	11.24	13.62
	+	+				76.01	11.24	13.62
	+	+	+			80.64	11.71	13.82
	+	+		+		83.67	11.74	14.08
	+	+		+	+	84.26	11.81	14.15

It can be seen that almost every strategy improved performance. Most notably, the biggest increase in performance can be seen between FPN and FPN++ (+7.18) and an accumulative increase of +8.14 when comparing our final method with FPN++. Comparing FPN++ and FPN++ with ECA, we chose to include ECA in our final model because of slightly better performance in 3- and 5-meter range. The benefit of MuRDE block is seen, as it gives an increase in performance by 7.57 when comparing it to FPN++ and ECA configurations. Finally, the choice of using DCNv2 compared to standard convolution is apparent, as MuRDE outperforms MuRE by 3.03 RDS points. Meter accuracies can be seen in Figure 10, and the same tendencies can be seen there as well. Notably, FAM feature fusion strategy, while it did increase RDS by only 0.59 points; improvements can be specifically seen in precision range (3-10 meters).

**Figure 10.** MA@k metric with different network configurations.

Additionally, we visualize the impact of different methods on the UL14 dataset in Figure 11. In said Figure, first line showcases the heatmap on the whole satellite image, and the second line—a zoomed-in view. We specifically visualize those cases, where RDS improvements were observed (compared to FPN++, ECA configuration). In those cases, MuRDE and FAM blocks can be seen to be beneficial not only in reducing overall distance between predicted and ground truth locations, but also in concentrating the predicted area.

To investigate how kernels of different sizes and dilations (and therefore—kernels of different receptive fields) affect the performance, we also conducted additional analysis. In this ablation experiment, we limited ourselves with kernels of sizes 3 and 5, given that final model should be as light as possible and kernels of bigger size are not as computationally effective as preferred.

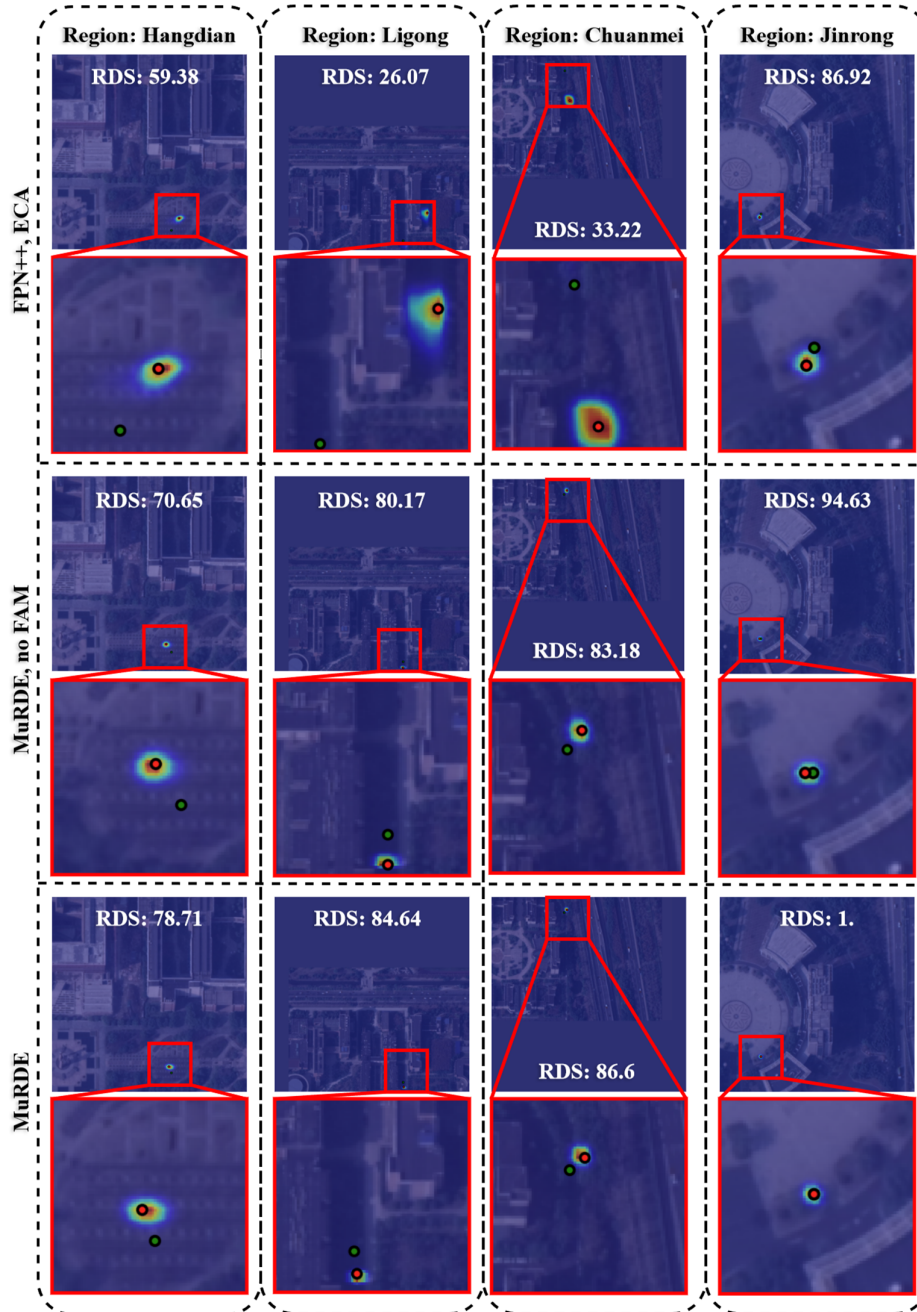


Figure 11. Heatmap visualization. Green dot denotes true location, red—predicted. RDS denotes RDS for the specific satellite map visualized.

Results are shown in Table 5, and we use the following abbreviations: SDK denoting single deformable kernel of size 3 and dilation 1; DDK-A denoting double deformable kernel of sizes 3 and 5 and dilations 1 and 1 respectively; DDK-B denoting double deformable kernel of sizes 3 and 5 and dilations 1 and 2; TDK denotes triple kernel consisting of size 3, 5, 5 convolutions and 1, 1, 2 dilations. In such a way we constructed 4 different receptive field MuRDE modules: size 3 receptive field (SDK), size 3 and 5 receptive fields (DDK-A), size 3 and 9 receptive fields (DDK-B) and size 3, 5, 9 receptive field modules (TDK, our final model configuration).

Figure 12 visualizes meter metrics for MuRDE configuration, and it can be seen that just like the increase in RDS, these metrics increased as well when comparing single/double and triple kernels.

Table 5. Comparison of MuRDE module configuration impact. SK denotes single kernel, DK denotes dual kernel, TK denotes triple kernel.

Method				RDS	GFLOPs	Params (M)
SDK	DDK-A	DDK-B	TDK			
+				81.96	11.26	13.7
	+			83.30	11.54	13.92
		+		83.27	11.54	13.92
			+	84.26	11.81	14.15

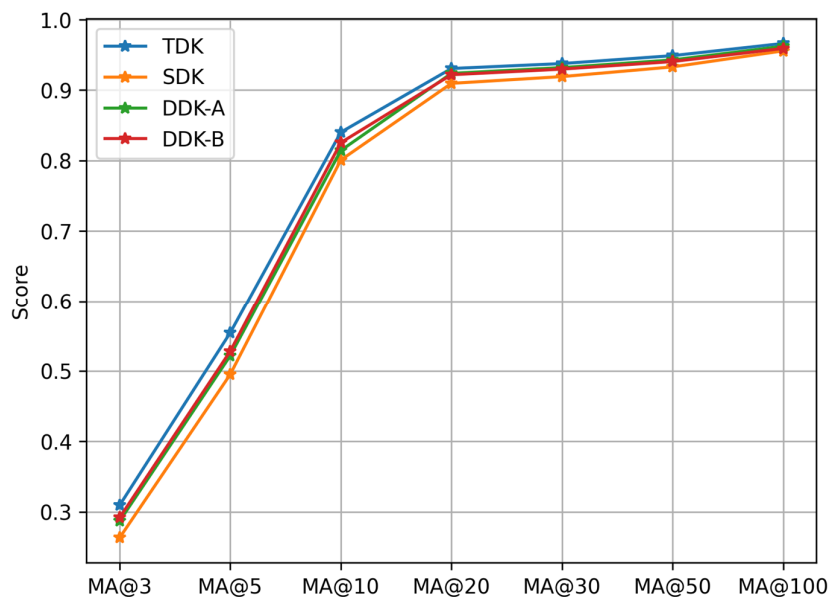


Figure 12. MA@k metric using different MuRDE kernel configurations.

7. Discussion

In our work, we treated the FPI task as a mixture of target tracking and segmentation, as on one hand, finding point in an image is targeting, but on the other hand, precision positioning should restore the final map as a dense heatmap. To enrich precise UAV positioning literature, we propose our MuRDE-FPN model, which utilizes a one-stream backbone with a custom designed FPN decoder. Our model architecture, MuRDE-FPN, addresses key challenges of multi-scale feature enrichment and spatial misalignment. By introducing the MultiReceptive Deformable Enhancement (MuRDE) module at the deepest stage of the feature pyramid, we enhance high-level semantic representations, and the Feature Alignment Module (FAM) adaptively merges the enriched features while preserving spatial consistency. We believe that our findings indicate the potential of custom FPN refining methods in the FPI task, with a special focus on specific scales, as with our method we observed improved performance in UAV localization in two datasets: UL14 and UAV-Sat.

We are, however, limited by our backbone design, and for this reason, our implications can be extended only to a specific setting: one-stream hierarchical backbone designs, in which cross-modal modeling is done at every stage. Even then, given the improvements on performance, some of our design choices can be integrated into different backbones. For instance, [14] noticed that effective usage of last feature significantly enhanced performance compared to other methods. Our findings are very similar, as we note an increase in performance when enhancing the most semantically rich layer as well. Notably, we adopt a multireceptive approach, and we see a performance increase when using non-deformable convolution as well. However, it is still unclear what could be the magnitude of improvement if such an approach would be integrated into other backbones, as [14–16] noticed that backbone design by itself impact the performance as well.

Model size and memory requirements matter a lot in UAV applications since on-board deployment dictates scarcity of resources. While our method is much lighter in terms of model size than the compared two-branch methods, some achieved similar results on the UL14 with lower computational cost [14–16]. In the future, we will address this issue with the smallest decrease in performance possible. In addition to popular model design solutions (model pruning techniques) to reduce the computational overhead, we will also aim to investigate different training strategies (knowledge distillation, deep supervision, contrastive pretraining) for optimal performance.

We also proposed a new dataset for proper evaluation of DL-based FPI methods. While similar datasets (derived from UAV-VisLoc) were proposed before [14,16,18], our derivation is unique in that the validation set is similar in its difficulty compared to the UL14 dataset and enables evaluation on much bigger satellite images. The structure of our validation set also allows us to evaluate the performance in the context of differing satellite search map sizes. There are, however, some gaps both in current literature and in our work: it is not clear how the end-to-end methods perform overall under different quality data. In particular, the gap exists when evaluating satellite images with visible/substantial change, e.g., in areas that experience rapid urbanization, destruction, climate change, or seasonal variation. In the future, we will aim to outline the important conditions under which the performance decrease could be suboptimal and to investigate data expansion and training techniques that inhibit said performance decrease.

Author Contributions: Conceptualization, M.K. and I.D.; methodology, M.K.; software, M.K.; validation, M.K.; formal analysis, M.K.; investigation, M.K.; resources, I.D.; data curation, M.K.; writing—original draft preparation, M.K.; writing—review and editing, M.K. and I.D.; visualization, M.K.; supervision, I.D.; project administration, I.D.; funding acquisition, I.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: Some of the data presented in this study can be made available upon request. For further information, please contact the corresponding author.

Acknowledgments: Authors would like to thank Vladislav Kolupayev for his insightful comments and discussions regarding overall scope of deep learning based approaches, which helped to improve our manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Manoj, H.M.; Shanthi, D.L.; Lakshmi, B.N.; Archana, K.J.; Venkata Naga Jyothi, E.; Archana, K. AI-Driven Drone Technology and Computer Vision for Early Detection of Crop Disease in Large Agricultural Areas. *Sci Rep* 2025, doi:10.1038/s41598-025-32384-1.
2. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* 2019, 7, 48572–48634.
3. Shayea, I.; Dushi, P.; Banafaa, M.; Rashid, R.A.; Ali, S.; Sarijari, M.A.; Daradkeh, Y.I.; Mohamad, H. Handover Management for Drones in Future Mobile Networks—A Survey. *Sensors* 2022, 22.
4. Zheng, T.; Xu, A.; Xu, X.; Liu, M. Modeling and Compensation of Inertial Sensor Errors in Measurement Systems. *Electronics (Switzerland)* 2023, 12, doi:10.3390/electronics12112458.
5. Lowe, D. Object Recognition from Local Scale-Invariant Features. 1999, 1150, doi:10.1109/iccv.1999.790410.
6. Liu, Y.; Wang, Y.; Wang, D.; Wu, W.; Li, X.X.; Sun, W.; Ren, X.; Song, H. A Scalable Benchmark to Evaluate the Robustness of Image Stitching under Simulated Distortions. *Sci Rep* 2025, 15, doi:10.1038/s41598-025-17730-7.
7. Sarlin, P.-E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning Feature Matching with Graph Neural Networks. 2020.

8. Gong, F.; Hao, J.; Du, C.; Wang, H.; Zhao, Y.; Yu, Y.; Ji, X. FIM-JFF: Lightweight and Fine-Grained Visual UAV Localization Algorithms in Complex Urban Electromagnetic Environments. *Information* 2025, *16*, 452, doi:10.3390/info16060452.
9. Ding, L.; Zhou, J.; Meng, L.; Long, Z. A Practical Cross-view Image Matching Method between Uav and Satellite for Uav-based Geo-localization. *Remote Sens (Basel)* 2021, *13*, 1–22, doi:10.3390/rs13010047.
10. Cui, Z.; Zhou, P.; Wang, X.; Zhang, Z.; Li, Y.; Li, H.; Zhang, Y. A Novel Geo-Localization Method for UAV and Satellite Images Using Cross-View Consistent Attention. *Remote Sens (Basel)* 2023, *15*, doi:10.3390/rs15194667.
11. Xu, Y.; Dai, M.; Cai, W.; Yang, W. Precise GPS-Denied UAV Self-Positioning via Context-Enhanced Cross-View Geo-Localization. *arXiv (Cornell University)* 2025, doi:10.48550/arxiv.2502.11408.
12. Dai, M.; Chen, J.; Lu, Y.; Hao, W.; Zheng, E. Finding Point with Image: An End-to-End for Vision-Based UAV Localization. 2022.
13. Chen, J.; Zheng, E.; Dai, M.; Chen, Y.; Lu, Y. OS-FPI: A Coarse-to-Fine One-Stream Network for UAV Geolocalization. *IEEE J Sel Top Appl Earth Obs Remote Sens* 2024, *17*, 7852–7866, doi:10.1109/JSTARS.2024.3380902.
14. Chen, N.; Fan, J.; Yuan, J.; Zheng, E. OBTPN: A Vision-Based Network for UAV Geo-Localization in Multi-Altitude Environments. *Drones* 2025, *9*, doi:10.3390/drones9010033.
15. Fan, J.; Zheng, E.; He, Y.; Yang, J. A Cross-View Geo-Localization Algorithm Using UAV Image and Satellite Image. *Sensors* 2024, *24*, doi:10.3390/s24123719.
16. Ju, C.; Xu, W.; Chen, N.; Zheng, E. An Efficient Pyramid Transformer Network for Cross-View Geo-Localization in Complex Terrains. *Drones* 2025, *9*, doi:10.3390/drones9050379.
17. He, Y.; Chen, F.; Chen, J.; Fan, J.; Zheng, E. DCD-FPI: A Deformable Convolution-Based Fusion Network for Unmanned Aerial Vehicle Localization. *IEEE Access* 2024, *12*, 129308–129318, doi:10.1109/ACCESS.2024.3415822.
18. Tian, L.; Shen, Q.; Gao, Y.; Wang, S.; Liu, Y.; Deng, Z. A Cross-Mamba Interaction Network for UAV-to-Satellite Geolocalization. *Drones* 2025, *9*, 427, doi:10.3390/drones9060427.
19. Yao, Y.; Sun, C.; Wang, T.; Yang, J.; Zheng, E. UAV Geo-Localization Dataset and Method Based on Cross-View Matching. *Sensors* 2024, *24*, doi:10.3390/s24216905.
20. Vaswani, A.; Brain, G.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. *Attention Is All You Need*; 2017;
21. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2021.
22. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. 2021.
23. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. 2021.
24. Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; Shen, C. Twins: Revisiting the Design of Spatial Attention in Vision Transformers. *arXiv (Cornell University)* 2021, doi:10.48550/arxiv.2104.13840.
25. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. PVT v2: Improved Baselines with Pyramid Vision Transformer. 2023, doi:10.1007/s41095-022-0274-8.
26. Wang, Q.; Zhang, J.; Yang, K.; Peng, K.; Stiefelhagen, R. MatchFormer: Interleaving Attention in Transformers for Feature Matching. 2022. *arXiv (Cornell University)* 2022, doi:10.48550/arxiv.2203.09645.
27. Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; Zhou, X. LoFTR: Detector-Free Local Feature Matching with Transformers. 2021, 8918, doi:10.1109/cvpr46437.2021.00881.
28. Ye, B.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework. 2022; p. 341.
29. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. 2017, 936, doi:10.1109/cvpr.2017.106.
30. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. 2018, doi:10.48550/arxiv.1803.01534.

31. Ghiasi, G.; Lin, T.-Y.; Pang, R.; Le, Q. V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. 2019 , 7029, doi:10.1109/cvpr.2019.00720.
32. Tan, M.; Pang, R.; Le, Q. V. EfficientDet: Scalable and Efficient Object Detection. 2020, 10778, doi:10.1109/cvpr42600.2020.01079.
33. Sun, G.; Jiang, X.; Lin, W. DBEENet: Dual-Branch Edge-Enhanced Network for Semantic Segmentation of USV Maritime Images. *Ocean Engineering* 2025, 341, doi:10.1016/j.oceaneng.2025.122731.
34. Xie, C.; Li, M.; Zeng, H.; Luo, J.; Zhang, L. MaSS13K: A Matting-Level Semantic Segmentation Benchmark. arXiv (Cornell University) 2025.
35. Thuan, N.H.; Oanh, N.T.; Thuy, N.T.; Perry, S.; Sang, D.V. RaBiT: An Efficient Transformer Using Bidirectional Feature Pyramid Network with Reverse Attention for Colon Polyp Segmentation. 2023. <https://doi.org/10.48550/arXiv.2307.06420>
36. Zhang, R.; Xie, M.; Liu, Q. CFRA-Net: Fusing Coarse-to-Fine Refinement and Reverse Attention for Lesion Segmentation in Medical Images. *Biomed Signal Process Control* 2025, 109, doi:10.1016/j.bspc.2025.107997.
37. Zhou, G.; Xu, Q.; Liu, Y.; Liu, Q.; Ren, A.; Zhou, X.; Li, H.; Shen, J. Lightweight Multiscale Feature Fusion and Multireceptive Field Feature Enhancement for Small Object Detection in the Aerial Images. *IEEE Transactions on Geoscience and Remote Sensing* 2025, 63, doi:10.1109/TGRS.2025.3602640.
38. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training Data-Efficient Image Transformers & Distillation through Attention. 2020, doi:10.48550/arxiv.2012.12877.
39. Wang, G.; Chen, J.; Dai, M.; Zheng, E. WAMF-FPI: A Weight-Adaptive Multi-Feature Fusion Network for UAV Localization. *Remote Sens (Basel)* 2023, 15, doi:10.3390/rs15040910.
40. Xu, W.; Yao, Y.; Cao, J.; Wei, Z.; Liu, C.; Wang, J.; Peng, M. UAV-VisLoc: A Large-Scale Dataset for UAV Visual Localization. 2024, doi:10.48550/arxiv.2405.11936.
41. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; IEEE Computer Society, 2020; pp. 11531–11539 , doi:10.1109/cvpr42600.2020.01155.
42. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision; Institute of Electrical and Electronics Engineers Inc., December 22 2017; Vol. 2017-October, pp. 764–773 , doi:10.48550/arxiv.1811.11168.
43. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable ConvNets v2: More Deformable, Better Results. 2018, doi:10.48550/arxiv.1811.11168.
44. Dong, X.; Qin, Y.; Fu, R.; Gao, Y.; Liu, S.; Ye, Y.; Li, B. Multiscale Deformable Attention and Multilevel Features Aggregation for Remote Sensing Object Detection. *IEEE Geoscience and Remote Sensing Letters* 2022, 19, doi:10.1109/LGRS.2022.3178479.
45. Fu, X.; Yuan, Z.; Yu, T.; Ge, Y. DA-FPN: Deformable Convolution and Feature Alignment for Object Detection. *Electronics (Switzerland)* 2023, 12, doi:10.3390/electronics12061354.
46. Huang, S.; Lu, Z.; Cheng, R.; He, C. FaPN: Feature-Aligned Pyramid Network for Dense Image Prediction; In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); October 1 2021; p. 844.
47. Li, J.; Wang, Q.; Dong, H. BAFFN: Bidirectionally Aligning Features to Improve Object Localization Accuracy in Remote Sensing Images. *Applied Intelligence* 2025, 55, doi:10.1007/s10489-025-06852-0.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.