Transformers For Recognition In Overhead Imagery: A Reality Check

Francesco Luzi
Duke University
francesco.luzi@duke.edu

Aneesh Gupta
Duke University
aneeshgupta8@gmail.com

Leslie Collins
Duke University
leslie.collins@duke.edu

Kyle Bradbury
Duke University
kyle.bradbury@duke.edu

Jordan Malof
University of Montana
jordan.malof@umontana.edu

Abstract

There is evidence that transformers offer state-of-the-art recognition performance on tasks involving overhead imagery (e.g., satellite imagery). However, it is difficult to make unbiased empirical comparisons between competing deep learning models, making it unclear whether, and to what extent, transformer-based models are beneficial. In this paper we systematically compare the impact of adding transformer structures into state-of-the-art segmentation models for overhead imagery. Each model is given a similar budget of free parameters, and their hyperparameters are optimized using Bayesian Optimization with a fixed quantity of data and computation time. We conduct our experiments with a large and diverse dataset comprising two large public benchmarks: Inria and DeepGlobe. We perform additional ablation studies to explore the impact of specific transformer-based modeling choices. Our results suggest that transformers provide consistent, but modest, performance improvements. We only observe this advantage however in hybrid models that combine convolutional and transformer-based structures, while fully transformer-based models achieve relatively poor performance.

1. Introduction

Transformer-based models have become prevalent in computer vision tasks and have achieved state-of-the-art performance in classification [12, 31], object detection [10, 48], and segmentation [7, 3]. This success might ostensibly suggest that transformers are superior to other existing models, such as those based upon convolutional structures, however this is difficult to conclude based upon the existing research literature due to the absence of experimental controls when comparing different vision models. The performance of modern vision models - all of which are based upon deep neural networks - are affected by numerous factors that vary widely among competing models used in public benchmarks, and in the research literature [34]. This includes factors such as the quantity and quality of training data, the training algorithm (e.g., optimizer), the training time allotted, and the model's size (i.e., the number of free model parameters). Another more subtle, but highly influential factor, is the computation time and effort invested by the designer on hyperparameter optimization, which can result in misleading performance comparisons [34].

Dataset	Region	Country	Size (km^2)
Inria	Austin	USA	81
	Chicago	USA	81
	Kitsap County	USA	81
	West Tyrol	Austria	81
	Vienna	Austria	81
DeepGlobe	Las Vegas	USA	150.2
	Paris	France	41.88
	Shanghai	China	173.32
	Khartoum	Sudan	32.88

Table 1. The cities and their size that compose the Inria and DeepGlobe datasets.

If one or more of the aforementioned factors vary between competing vision models, then it is unclear which factors among them are responsible for any performance differences [38, 24, 20, 34]. Consequently, it is unclear whether the recent success of transformer-based models applied to overhead imagery has been driven by the use of transformers, or the variety of other factors that vary among the competing models. A major goal of vision research is to uncover the underlying causal factors and design principles that underpin vision systems; this not only advances our understanding of vision systems, but also often leads to substantive performance improvements in such systems. Therefore an important question in the vision literature is whether, and to what extent, transformers generally benefit vision models. Controlled studies of transformers have been conducted with natural imagery [37], providing some

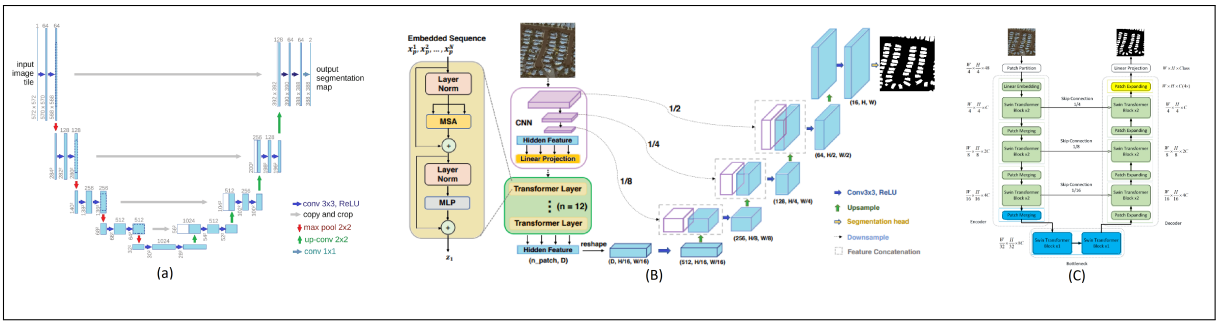

Figure 1. (a), (b), and (c) show the architecture for the Unet, TransUnet, and SwinUnet respectively. These figures were inspired directly by [40, 7, 3], respectively

evidence in that context. However, it is unclear whether their success extends to the unique statistics and conditions present in overhead imagery, a major area of vision research. Transformers excel at modeling long range dependencies, which while generally beneficial in most vision tasks, may not be as important in segmentation of overhead imagery where building information is compact, highly localized, and many times isolated from other structures. To our knowledge there has been no systematic study of this question for overhead imagery tasks.

In this work we perform a carefully-controlled empirical comparison of three state-of-the-art segmentation models using overhead imagery, where each model utilizes progressively more transformer-based structures. Specifically, we consider the following three models: U-Net [21], TransUNet[7], and SwinUnet[3]. This is the first time the TransUnet and SwinUnet have been applied to a large-scale dataset of overhead imagery [1]. Aside from the model variation we carefully control all other experimental factors, such as the size of the models, their quantity of training data, and training procedures. We use a large and diverse dataset of overhead imagery, comprising two publicly-available benchmarks to maximize the generality and relevance of our results. To provide a transparent and unbiased hyperparameter optimization procedure, we use Bayesian Optimization (BO) with a fixed budget of iterations to select the hyperparameters of each model. We provide each model with approximately 330 hours of optimization time in order to identify effective hyperparameters for each model. These experimental controls allow us to study whether, and to what degree, transformers are beneficial in the context of overhead imagery. Using our optimized models, we also conduct several additional ablation studies to evaluate the impact of specific design choices in the transformer-based models. We can summarize our contributions as follows:

- The first investigation of two recent state-of-the-art segmentation models for processing overhead imagery: the transUnet [7], and the swinUnet [3].
- The first controlled evaluation of whether, and to what extent, transformers are beneficial for vision models in overhead imagery.

[1]The recent work in [43] independently and concurrently studied these models, in a complementary setting

2. Related Work

Segmentation in overhead imagery. Segmentation of overhead imagery requires complex features to describe the vast domain as well as pixel level precision. Initially developed for medical imagery, Unet [40] has been show to be a powerful model in overhead image segmentation [18, 16] and in the broader segmentation community, with many variations on the model such as Dense-Unet [29], Res-Unet [44], Unet++ [50], V-Net[33], and Unet3+ [19]. This is attributed to the auto encoder-like structure where it receives its ”U” shape and name combined with the skip connections, feeding high resolution spatial information into the last layers of the model.

Other models such as DeepLabv3 [9] and Mask-RCNN [14] have been used successfully in segmentation of overhead imagery [27, 4]. While these models also perform very well, we chose to evaluate Unet-based architectures due to the high performance and large number of variant models. This high number of variants allowed us to more easily compare small changes in the model architecture.

Transformers in segmentation. Very recently transformers have started to be used in that segmentation of overhead imagery [17, 42], achieving good performance. Transformers had already started to become common in other domains such as TransUnet [7], ViT-V-Net [6], TransClaw U-Net [5], UTNet [13], Cotr [45], and SwinUnet [3] in medical image segmentation.

Evaluation of transformers. Transformers are rather new in computer vision and have only recently become state-of-the-art. As a result, their impact on performance has not been thoroughly analyzed in many domains and applications, including our own. While work has been done in evaluating their generalization capabilities in respect to distribution shift [47] and how transferable their learned representations are [49], these are very general results about the feature representations derived for other applications. For segmentation of overhead imagery, to our knowledge, there

has been no work done towards thoroughly evaluating and isolating the effect of transformers in state-of-the-art models.

3. Benchmark datasets

We train and evaluate our data on two large publicly-available datasets of overhead imagery: the DeepGlobe (DG) Competition Dataset [11], and the Inria Building Labeling Competition Dataset [32, 18]. Both of these datasets contain high-resolution (0.3m ground sampling density) color imagery with pixel-wise labels indicating the presence of a building (a value of one), or not (a value of zero). Collectively, these datasets encompass nine diverse cities spanning North America, Europe, and Asia, as summarized in Table 1.

4. Benchmark Segmentation Models

Our aim is to compare similar state-of-the-art models, shown in Fig 1, and determine what factors contribute to the overall performance. Specifically, we aim to answer whether or not transformer layers are valuable in overhead segmentation. For this we selected the Unet [40], TransUnet [7], and SwinUnet [3] models since they all use the same base Unet structure, they are all state-of-of-the-art models in segmentation, and they contain different levels of transformer integration. SwinUnet being entirely transformer based, Unet being completely convolutional, and TransUnet a hybrid of transformer and convolutional neural networks.

For most of our comparisons we restrict the number of parameters for each model such that they can be compared fairly. The Unet and TransUnet natively have a parameter count around 105 million and so we select this for our baseline models. A brief description of each model and their importance is given below, a more detailed description can be found in the supplemental material.

Unet. For our baseline Convolutional model we used a Unet based architecture with a ResNet101 [15] backbone, shown in Fig 1 (a). The model uses ResNet101 weights pretrained on ImageNet [28] with the standard Unet structure described in the paper. We chose to use the Unet due to its popularity and performance in segmentation tasks [46, 1, 18, 22]. Its simple and intuitive design has allowed for many variants, including the transformer based variants we considered.

TransUnet. For the majority of our experiments we use a standard TransUnet with Visual Transformer (ViT) blocks [12] pretrained on ImageNet, shown in Fig 1 (b). The official implementation is used for the model, and all model changes are derived from that code base. The TransUnet proved to be an excellent choice for a convolution-transformer mixed model due to its simple modification. The TransUnet is very similar to the standard Unet but with transformer layers in the deepest part of the encoder. This allows for an ablation over different aspects of the model without effect on the rest of the layers.

Parameter	Range
Learning Rate	10^{μ}, $\mu \in U(-4, -1)$
Weight Decay	10^{μ}, $\mu \in U(-7, -3)$
Window Size	[2, 4, 8]

Table 2. Here we list the parameters used in BO in the first column and their possible values in the second column. μ is taken from a uniform distribution. Learning rate and weight decay are found using Bayesian optimization in all models and window size is only used in the Bayesian optimization for the SwinUnet.

SwinUnet. Our fully transformer-based model is a modified SwinUnet, shown in Fig 1 (c), with weights loaded from a Swin Base model [31]. We used the implementation from the original code base for all of our work. We had modified the original SwinUnet architecture to achieve an equivalent parameter count to the other models. We also train a SwinUnet with a Swin Tiny backbone architecture as is described in the SwinUnet paper and compare that to a Unet with equivalent parameters to verify that our modified SwinUnet is representative of the model's performance. We use the SwinUnet to represent fully transformer based models due to its state-of-the-art performance in medical segmentation and the Swin transformers dominance in other computer vision based tasks such as classification and object detection [31, 10].

5. Experimental Design

The primary goal of our study is to compare the effectiveness of recent transformer-based models to state-of-the-art convolutional models, while controlling the number of trainable model parameters.

5.1. Data Handling

While the DG dataset contained labels for multiple classes (e.g., road, building, etc.), we only used the building labels so that we can train on both datasets together. Our combined dataset involves a two thirds, one sixth, one sixth split for training, validation, and testing respectively. For Inria, we use the official test set as our test set (first six tiles) and for DG we randomly select one sixth of the data for the test set. Each city has the same proportions represented in the training, validation, and test sets.

5.2. Model hyperparameter optimization

To minimize bias towards a particular model, we optimized all competing models using BO [41], which is a systematic, replicable, and transparent process to search for optimal hyperparameters via experimentation. Furthermore, each model was allowed 30 iterations of BO, ensuring comparable computational resources were provided to each model. We chose to use 30 iterations based on our experiments with the Unet. We found that 30 trials explored the

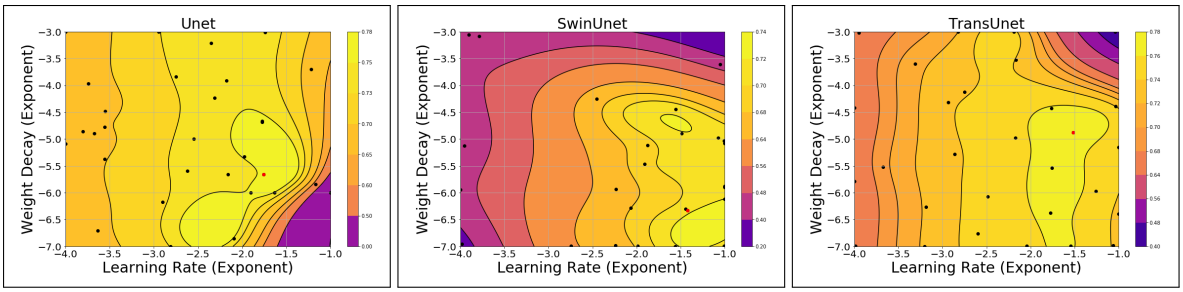

Figure 2. Heat maps of the parameter space searched with Bayesian Optimization. Sampled points are displayed as block dots, with the final parameter choice represented by a red dot, and Gaussian processes are used to model the points in between to fill out the space. Learning rate is plotted on the X-axis with weight decay plotted on the Y-axis. Both learning rate and weight decay are sampled by their exponential (-3 gives a value of 10^{-3} or $1e-3$) so that there is equal weight given to parameter values on a $\log_{10}$ scale.

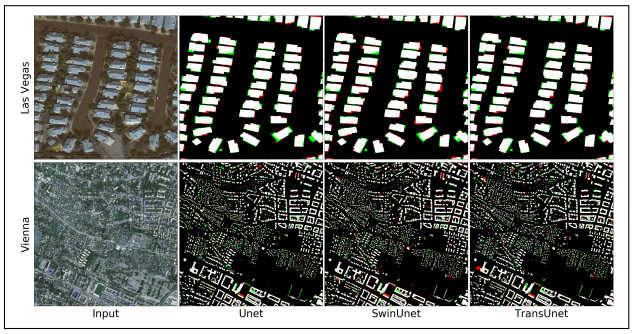

Figure 3. Examples of input images and mask outputs. Each row is a test image from a different city, with the first row from DG and the second from Inria. The first column contains the input image, the next three columns contain the predictions from the Unet, SwinUnet, and TransUnet respectively. The difference with the ground truth is highlighted, with green showing missed building pixels and red denoting false alarm building pixels.

hyperparameter space thoroughly and any more trials would give little to no performance improvements. After approximately the 20th iteration of BO, the models converged to a local minima and had minor to no improvements with continued search time. Fig 2 displays a predicted heat map of the parameter search space along with the points sampled during BO. In the supplemental work we include figures that show the model performance converging, training validation for each BO iteration, and the model performance corresponding to individual parameters selected. We also had limited computational resources and 30 trials took approximately two weeks to complete, requiring six weeks in total to optimize over the three models. For BO we used the python implementation provided by the BO library [36]. We initialized the BO with 2 random points, used the expected improvement [25] acquisition function and set the exploitation-exploration trade-off parameter ("xi") to 0.1, which we found to work well through initial trial and error. All other parameters were remained at their default values.

For each of our three model classes, we identified a small set (2-3) of the most influential hyperparameters for inclusion in the BO, shown in Table 2. This includes learning

Model	Inria	DG	Composite	Parameters (M)
Unet34	**76.58**	**78.71**	**77.74**	26.71
SwinUnet Tiny	75.67	77.85	76.87	27.13

Table 3. Smaller versions of Unet and SwinUnet were also trained to verify that the performance gap exists with the original SwinUnet architecture

rate and weight decay, with the addition of window size for the SwinUnet models. Our goal was to select the most impactful hyperparameters that would not change the number of parameters available to the model. In each case we trained and tested with randomly-selected hyperparameter settings to initialize the Gaussian Process in the BO. The models were trained using one fifth of the training dataset to expedite the process. Note that this still resulted in a relatively large training dataset, including 107 km^2 of satellite imagery spanning 9 regions, and requiring 12 hours per model trained. The last epoch performance on the validation set was used as the target parameter to optimize in the BO. The hyperparameters with the highest validation performance were then adopted to train a model on the full training dataset, which was then evaluated on the withheld test set as a final unbiased estimator of the model's performance.

The transformer-based models have additional impactful hyperparameters, such as the number of heads, embedding dimension, and layer count, however as discussed above, these can alter the number of trainable parameters in the model and therefore we fixed these parameters in advance. We kept these parameters consistent with what was used in the pretrained models whose weights we use to initialize our models. To avoid disadvantaging any model, we did not include these parameters in the BO. However, after performing BO, we study the impact of the number of transformer layers in a model by searching over a grid of settings. We do not explore these for the other architectural parameters since it would remove the ability to use pretrained weights.

Model	Inria					DeepGlobe				Composite	Parameters (M)
	Austin	Chicago	Kitsap County	West Tyrol	Vienna	Las Vegas	Paris	Shanghai	Khartoum		
Unet [40]	81.92	71.58	69.00	80.44	82.53	**85.50**	72.26	77.13	73.64	79.53	104.89
SwinUnet [3]	80.21	69.62	68.70	80.33	81.85	84.87	70.57	76.25	72.36	78.48	102.64
TransUnet [7]	**81.94**	**73.21**	**69.19**	**81.46**	**82.94**	85.45	**72.92**	**77.31**	**73.79**	**79.96**	105.91

Table 4. Comparison of performance of the three model architectures we tested broken down by city, measured by the intersection-over-union (IoU). The Unet represents the convolutional only approach, SwinUnet the transformer only approach, and TransUnet being a combination of the two. Model architecture was selected so that the models would have roughly similar parameter counts. The best performance on each city is shown in bold.

5.3. Training

All models were trained using the hyperparameters found from BO on the base model unless stated otherwise. For example the TransUnet with 6 transformer layers used the same parameters as were found using the TransUnet with 12 layers. This approach was used to decrease search time, and we observed that hyperparameters shared between similar architectures perform very well. Since training on a subset of the data was utilized to save time, we used an adaptive learning schedule and training time. This avoided the issue of selecting a learning schedule that may have unfairly advantaged one model over another, as well as making the transition from utilizing a subset of the training data set to utilizing the full data set smoother. We allowed the models to train until the validation curve flattened and then reduced the learning rate by a factor of 2. The learning rate was dropped three times and then the training was halted. To determine if the validation had flattened we maintained a running average of the validation performance and compared the current value against the running average obtained 10 epochs prior. Once there was zero difference between the two, or the difference was negative, the learning rate was lowered. After the third drop in learning rate, training was continued until the validation curve flattened once more and then training was halted with the last epoch validation used for scoring the parameter selection.

All models were trained, validated, and tested on Inria and DG. 650×650 patches were taken from each satellite image, normalized by the dataset statistics, and cropped randomly during training to be 512×512. Random rotations of $90°$ were also used during training. The models were evaluated on a pixel-wise cross entropy loss and a soft intersection over union (IoU) [39] loss, weighted equally.

5.4. Performance Metrics

We report performance using intersection-over-union (IoU), because it is widely-used for segmentation of overhead imagery, and is the official performance metric for the Inria and DG datasets [11, 32]. IoU measures the intersection of all predicted building pixels and ground truth labels over the union of all predicted building pixels and ground truth labels. The IoU is given by,

$$IoU = \frac{Prediction \cap Labels}{Prediction \cup labels} \quad (1)$$

This results is a metric normalized between 0 and 1 that intuitively encapsulates how well the model predicted the ground truth labels. The three right columns of images in Fig 3 demonstrate an example of prediction vs. ground truth where the intersection is shown by the white pixels and the union is all non-black pixels.

6. Model Performance Comparisons

We performed a number of experiments to isolate specific changes and impact of the model and parameter selection process. We evaluated the effect that different numbers of transformer layers, pretraining, parameter searches, and transformer layers in general had on the model's performance. Fig 3 demonstrates each model's performance on our test sets.

Standard-sized models. The results from our standard-sized models are reported in Table 4, where TransUnet achieves the highest overall IoU, followed by the Unet and the SwinUnet, respectively. These rankings persist for each of the two benchmark datasets as well, suggesting that the results are somewhat robust to variations in the underlying data. These results are comparable to current state-of-the-art results on Inria [8, 26, 51]. The results *tentatively* suggest that including some transformer modules (e.g., the TransUnet) is beneficial, however including too much can be detrimental (e.g., SwinUnet). The TransUnet performs consistently better than the Unet, which reflects the finding in the TransUnet and SwinUnet papers [7, 3], but the SwinUnet results seem to under-perform. Prior research has found that Transformer-based models are more difficult to train than convolutional models, and tend to improve faster with growing quantities of training data [30, 12]. It is possible therefore that, as overhead imagery datasets continue to grow, the SwinUnet may perform relatively better. Given the sample complexity and size of our training data however, it does not. The lower performance of SwinUnet in this setting provides support to the notion that long range dependencies are not as beneficial in the early stages of segmentation in overhead imagery and that localized information is important in overhead imagery.

Small-sized models. To fairly compare the SwinUnet model and increase its parameter count we added transformer blocks and loaded the model from the pretrained Swin Base checkpoint, using the base architecture instead

(e.g. number of heads, hidden layer dimensions). This could disadvantage the model since we have not performed an extensive architectural search for the optimal large SwinUnet, whereas, the Unet and TransUnet were tested in their original form and thus were likely more optimized. To verify that in enlarging the SwinUnet architecture we didn't disadvantage the model in training we also compared the SwinUnet with a Swin Tiny backbone to the Unet with ResNet34 structure. We performed BO on both these models in the same manner as described above and trained them in exactly same manner. Table 3 shows the results on the test set and the results are consistent with the results found using the larger models as reported in Table 4.

6.1. Are Transformers Beneficial?

Under fair conditions TransUnet, a transformer-convolutional hybrid, out-performed both the alternate Unet and SwinUnet on a large and diverse test set. While this gives credibility to the notion that transformers improve a model's performance for segmentation of overhead imagery, this claim cannot be made based on the performance improvements alone. We have found that while removing all transformer layers from TransUnet decreases performance, as shown in Table 6, the model still outperforms the small Unet and performs similarly to the large Unet. Since the TransUnet with no transformer layers and a greatly reduced model size (by parameter count) performs on a similar level to the Unet then it is clear that there are other factors in the TransUnet architecture that boost performance. To resolve this ambiguity, we performed an ablation by substituting the transformer layers in the TransUnet with other layer types, shown in Table 7. We found that while the benefit is small, adding transformers can improve model performance over other layer types.

7. Additional Analysis and Ablations

7.1. Window Size Effect

One of the parameters used for the SwinUnet in the BO search was the window size. We found in the search that a window size of 4 was ideal for performance but wanted to verify the accuracy of the search and the importance of this parameter. In Table 5 we provide the performance of 3 models trained with window size of 2, 4, and 8. The results show that the selected window size performed best and that increasing the window size had negligible effects while reducing it caused a large drop in performance.

7.2. Layer Ablation

We considered the effect that the number of transformer layers used had on performance. Table 6 shows the performance of the TransUnet with 0, 6, 8, 10, and 12 transformer layers. The Unet is included for comparison. We found

Model	Window Size	Inria	DG	Composite	Parameters (M)
SwinUnet	2	76.90	78.95	78.02	102.62
	4	**77.18**	**79.55**	**78.48**	102.64
	8	77.12	79.10	78.21	102.73

Table 5. We trained the SwinUnet with varying window sizes to determine the effect of window size on the model and to verify that the BO correctly picked the highest performing parameter. Each row provides the performance on the test set of a SwinUnet model trained with a different window size. All other training parameters were kept constant.

Model	Transformer Layers	Inria	DG	Composite	Parameters (M)
Unet34	0	76.58	78.71	77.74	26.71
Unet101	0	78.41	80.46	79.53	104.89
TransUnet	0	78.65	80.20	79.51	20.86
	6	80.02	80.83	80.47	63.38
	8	**80.12**	81.04	**80.63**	77.56
	10	79.90	80.94	80.47	91.74
	12	79.22	80.56	79.96	105.91
	14	79.98	**81.07**	80.58	120.09

Table 6. We performed an ablation over the number of transformer layers used. Unet and TransUnet with 0 layers are included to show the benefit that transformer layers provide.

that on average the TransUnet with 8 transformer layers performed the best but that the performance did not change dramatically for any model other than the 0 layer TransUnet. We also found that the 12 layer TransUnet performed the worst of all of the TransUnet models with transformer layers. This leads us to believe that the BO search found generally good hyperparameters that work well across different variations of the TransUnet.

Another interesting finding was that the TransUnet with 0 transformer layers performed almost as well as it did with transformer layers; it outperformed the RestNet34-based U-Net despite it have somewhat more parameters, and it performed equivalent to the Unet baseline model despite it having substantially more parameters. This implies that the transformer portion of the TransUnet is only one factor in its performance advantage over the Unet architecture, and that there are other factors contributing to the TransUNet's performance advantages. One notable difference between the Unet and the TransUnet is that the Unet does not use padding in many of its convolutional layers, leading to smaller feature tensors and resulting in a difference in the dimensions between the encoder and decoder. This difference in dimensions does not allow for all of the encoder features to be passed to the decoder via the skip connections and reduces the amount of high resolution information provided to the decoder. While we did not test this hypothesis, which would require large architectural changes to either the TransUnet or the Unet, there were few other factors that we believe could cause such a large difference in perfor-

Model	Architecture Type	Pretraining Used	Inria	DG	Composite	Parameters (M)
TransUnet	Fully Connected	No	76.07	77.75	76.99	103.76
TransUnet	Convolutional	No	77.42	79.48	78.55	105.82
TransUnet	Transformer	No	77.88	79.66	78.86	105.91
TransUnet	Transformer	Yes	79.22	80.56	79.96	105.91

Table 7. We explore the effect of using transformer layers in the last stage of the encoder. Here we compare the standard TransUnet model with variations that replace the transformer layers with fully connected layers or 3 convolutional blocks. Note that pretrained weights are used for all layers except the transformer layers and their replacements.

Model	Hyperparameters Used	Inria	DG	Composite
TransUnet	TransUnet	79.22	80.56	79.96
	Unet	79.65	80.91	80.35
Unet	TransUnet	78.57	80.36	79.55
	Unet	78.41	80.46	79.53

Table 8. We tested the effects of using hyperparameters found from a BO search on one model when applied to another model. For example we used the hyperparameters found through our BO procedure on Unet and trained a TransUnet with those parameters. The table shows that performance is somewhat robust to the hyperparameters used for training. The "Hyperparameters Used" column denotes which model was used in the hyperparameter search.

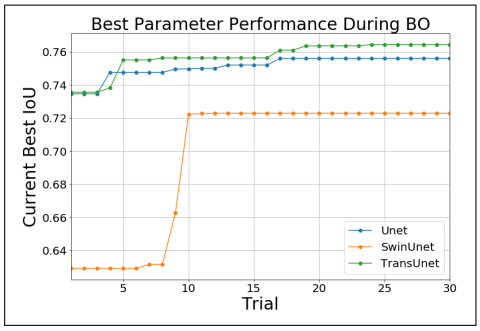

Figure 4. The max IoU trial versus the current iteration is displayed for all three baseline Bayesian optimizations. Each model improves greatly in the beginning but then converges at around or before the 20th iteration, with minor or no improvements in performance afterwards.

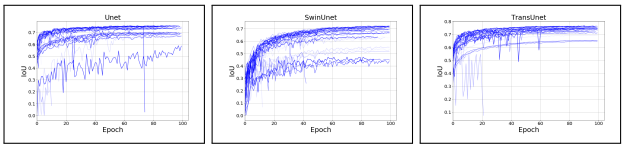

Figure 5. Shown here is the training validation at every epoch for each iteration of Bayesian optimization. Each model was allowed 30 iteration of Bayesian optimization to find the optimum hyperparameters. The darker lines represent the later trials and the lighter the line, the earlier the trial. Since we used an adaptive stopping criteria some trials end earlier if their performance stagnates or is very poor for that stage of training.

mance.

7.3. Pretraining and Parameter Effect

It is well known that pretraining is important for good model performance, this is especially important for transformer models [12]. We evaluated the impact that pretraining has by training a model with randomized weights for the transformer layers only. Table 7 displays the performance of the TransUnet with and without pretraining for its transformer layer. Note that pretrained weights were still used for all other layers of the model. The randomized TransUnet underperforms even the TransUnet with no transformer layer, indicating that randomizing the weights is more detrimental to the performance than not including the layers.

We also investigated the importance of using BO for each model architecture was. It is very time consuming to optimize for every architectural change in the model and cross architecture comparisons become moot if there is large variability in the performance depending on how the hyper parameters are obtained. To test this we use the hyperparameters found from the Unet BO search to train the TransUnet and vice versa. Table 8 shows that performance is consistent in this scenario and actually improves for both models.

7.4. Transformer Effect

In discovering that the transformer layer is not necessary for improvement over the baseline Unet, we explored some variations of the baseline TransUnet model by replacing the transformer layer with other layer types. Since these modifications are non-standard we did not have pretrained weights for these layers and thus randomly initialized them. To account for this, we compare the modified models to the TransUnet with pretrained and randomized weights. This is not a perfect comparison since transformer models seem to be more sensitive to pretraining in general.

First, we simply replaced the transformer layers with fully connected layers. The input to these fully connected layers were the same patch embedding used for the transformer layers. The authors of Unet motivate using transformer layers to help gather global context in the encoding process. Fully connected layers should be able to also model global relationships in the image. The added fully connected layers are of the same input and output dimensions as the input embedding. We used ReLU activation [35] and layer normalization [2] between every fully connected layer. Enough layers were added such that the total parameter count was similar to our standard TransUnet model.

We also evaluated the effect of replacing the transformer layers with more convolutional layers. We removed the linear embedding as to retain the spatial dimensions of the features. We did not explore many different configurations for the convolutional replacements and just used simple 3×3 convolutional filters with the same channel size as the transformer layers and residual connections between layers. We used ReLU activation and batch normalization [23] after each convolutional layer. To keep the comparisons as fair as possible we added enough layers to increase the parameter count to be equal to the standard TransUnet.

The results presented in Table 7 show that using transformer layers has a greater impact than naively using other layer types. While it is impossible to say how much larger of an impact transformers have on performance compared to using convolutional or fully connected layers without an exhaustive search over the architectural space, we have found that using transformers provided slight performance improvements over using convolutions or fully connected layers in a reasonable manner.

7.5. Effectiveness of Bayesian Optimization

The conclusions of this work depends heavily upon the premise that BO found good hyperparameters for each of our competing models, reflecting their performance in practice, given a typical systematic optimization of model hyperparameters. In this section we provide evidence that *three* key steps of our BO was effective, providing strong evidence the BO process as a whole were effective.

First we present evidence that BO found near-optimal parameters within its search range. To do this, for each of our three BOs (one for each competing model in our experiments), we report the model's performance (IOU) estimated by the Gaussian Process model as a function of hyperparameter settings. These estimates are made over a dense grid and are reported as an image in Fig. 2, where we have also overlaid the IoUs obtained by experiment at hyperparameter settings sampled by the BO. From these results, we see that there was a clear local optimum for each of the three models, suggesting that we chose sufficiently large search ranges for each model to find good hyperparameters. Furthermore, the BO sampled one (or more) points near to these local optima, suggesting that near-locally-optimal hyperparameters were obtained for each model.

These conclusions are corroborated by results presented in Fig. 4, where we also report the maximum IoU obtained as a function of the number of BO iterations run for each model. We see that for each model the IoUs initially found were relatively low, and then (often steadily) increased until reaching some point of saturation, where greater IoUs were not found after many iterations. It is clear from Fig 2 too that the BO models were not simply sampling similar settings (which is possible in BO with poor BO hyperparameter settings), but instead they sampled a diverse set of hyperparameters across the search space. These results suggest that BO effectively improved the hyperparameter settings, and did so until a robust optimum setting was found.

Collectively, these results suggest that BO was effective, as long as the IoUs obtained from individual experiments (i.e., training and validating a model with a single hyperparameter) were valid. This is not guaranteed, since we needed to carefully design an automatic stopping criteria for training the models, which allowed models to train until they consistently did not improve, and before any overfitting reduces their performance. In Fig. 5 we report the validation IOU as a function of epoch for all of the models trained during the BO process, where most models exhibit expected validation error during training, and appear to saturate in IOU at, or before, the end of training.

8. Conclusion

In this work we studied whether, and to what degree, transformers are beneficial for segmentation tasks in overhead imagery. To address this question, we performed a large-scale systematic empirical comparison of three state-of-the-art segmentation models, where each model utilizes progressively more transformer-based structures. We considered the following three models: U-Net [21], TransUnet [7], and SwinUnet [3]. Based upon our results, we make the following conclusions:

- Transformers provide consistent, but modest, performance improvements. This performance advantage was only observed in hybrid architectures (e.g., the TransUnet), comprising a convolutional encoder followed by transformers, which performed best among all models. Fully transformer-based models (e.g., SwinUnet) achieved relatively poor performance.
- We found that the U-Net structure used in the Trans-U-Net (e.g., when operated without any transformer layers) performed better than other U-Net structures of comparable size (see Section 7.2). We were unable to isolate the precise cause of this advantage.

To our knowledge, this represents the most systematic comparison of transformers within the remote sensing literature to date, providing more robust evidence regarding the impact of transformers. We note several limitations of our experiments that should be considered when interpreting our results: (i) we focused on segmentation tasks; (ii) we only employed building target classes; (iii) we used relatively small training sets compared to color imagery applications.

Acknowledgment

We thank the Energy Initiative at Duke University for their support. This work was supported in part by the Alfred P. Sloan Foundation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Alfred P. Sloan Foundation.

References

[1] Abdelilah Adiba, Hicham Hajji, and Mustapha Maatouk. Transfer learning and u-net for buildings segmentation. In *Proceedings of the New Challenges in Data Sciences: Acts of the Second Conference of the Moroccan Classification Society*, pages 1–6, 2019.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021.

[4] Osmar Luiz Ferreira de Carvalho, Osmar Abilio de Carvalho Junior, Anesmar Olino de Albuquerque, Pablo Pozzobon de Bem, Cristiano Rosa Silva, Pedro Henrique Guimaraes Ferreira, Rebeca dos Santos de Moura, Roberto Arnaldo Trancoso Gomes, Renato Fontes Guimaraes, and Dibio Leandro Borges. Instance segmentation for large, multi-channel remote sensing imagery using mask-rcnn and a mosaicking approach. *Remote Sensing*, 13(1):39, 2020.

[5] Yao Chang, Hu Menghan, Zhai Guangtao, and Zhang Xiao-Ping. Transclaw u-net: Claw u-net with transformers for medical image segmentation. *arXiv preprint arXiv:2107.05188*, 2021.

[6] Junyu Chen, Yufan He, Eric C Frey, Ye Li, and Yong Du. Vit-v-net: Vision transformer for unsupervised volumetric medical image registration. *arXiv preprint arXiv:2104.06468*, 2021.

[7] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[8] Keyan Chen, Zhengxia Zou, and Zhenwei Shi. Building extraction from remote sensing images with sparse token transformers. *Remote Sensing*, 13(21):4441, 2021.

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[10] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7373–7382, 2021.

[11] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 172–181, 2018.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Yunhe Gao, Mu Zhou, and Dimitris N Metaxas. Utnet: a hybrid transformer architecture for medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 61–71. Springer, 2021.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Nanjun He, Leyuan Fang, and Antonio Plaza. Hybrid first and second order attention unet for building segmentation in remote sensing images. *Science China Information Sciences*, 63(4):1–12, 2020.

[17] Xin He, Yong Zhou, Jiaqi Zhao, Di Zhang, Rui Yao, and Yong Xue. Swin transformer embedding unet for remote sensing image semantic segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022.

[18] Bohao Huang, Kangkang Lu, Nicolas Audeberr, Andrew Khalel, Yuliya Tarabalka, Jordan Malof, Alexandre Boulch, Bertr Le Saux, Leslie Collins, Kyle Bradbury, et al. Large-scale semantic classification: outcome of the first year of inria aerial image labeling benchmark. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 6947–6950. IEEE, 2018.

[19] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059. IEEE, 2020.

[20] Matthew Hutson. Artificial intelligence faces reproducibility crisis, 2018.

[21] Vladimir Iglovikov, Selim Seferbekov, Alexander Buslaev, and Alexey Shvets. Ternausnetv2: Fully convolutional network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[22] Vladimir Iglovikov and Alexey Shvets. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*, 2018.

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[24] Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *arXiv preprint arXiv:1708.04133*, 2017.

[25] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[26] Jian Kang, Zhirui Wang, Ruoxin Zhu, Xian Sun, Ruben Fernandez-Beltran, and Antonio Plaza. Picoco: Pixelwise contrast and consistency learning for semisupervised building footprint segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:10548–10559, 2021.

[27] Fanjie Kong, Bohao Huang, Kyle Bradbury, and Jordan Malof. The synthinel-1 dataset: A collection of high resolution synthetic overhead imagery for building segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1814–1823, 2020.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[29] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE transactions on medical imaging*, 37(12):2663–2674, 2018.

[30] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems*, 34:23818–23830, 2021.

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[32] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3226–3229. IEEE, 2017.

[33] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[34] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020.

[35] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[36] Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, 2014–.

[37] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022.

[38] Félix Renard, Soulaimane Guedria, Noel De Palma, and Nicolas Vuillerme. Variability and reproducibility in deep learning for medical image segmentation. *Scientific Reports*, 10(1):1–16, 2020.

[39] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[41] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[42] Zhongyu Sun, Wangping Zhou, Chen Ding, and Min Xia. Multi-resolution transformer network for building and road segmentation of remote sensing image. *ISPRS International Journal of Geo-Information*, 11(3):165, 2022.

[43] Libo Wang, Rui Li, Ce Zhang, Shenghui Fang, Chenxi Duan, Xiaoliang Meng, and Peter M Atkinson. Unetformer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 190:196–214, 2022.

[44] Xiao Xiao, Shen Lian, Zhiming Luo, and Shaozi Li. Weighted res-unet for high-quality retina vessel segmentation. In *2018 9th international conference on information technology in medicine and education (ITME)*, pages 327–331. IEEE, 2018.

[45] Yutong Xie, Jianpeng Zhang, Chunhua Shen, and Yong Xia. Cotr: Efficiently bridging cnn and transformer for 3d medical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 171–180. Springer, 2021.

[46] Huanran Ye, Sheng Liu, Kun Jin, and Haohao Cheng. Ct-unet: An improved neural network based on u-net for building segmentation in remote sensing images. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 166–172. IEEE, 2021.

[47] Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7277–7286, 2022.

[48] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[49] Hong-Yu Zhou, Chixiang Lu, Sibei Yang, and Yizhou Yu. Convnets vs. transformers: Whose visual representations are more transferable? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2230–2238, 2021.

[50] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018.

[51] Stefano Zorzi, Ksenia Bittner, and Friedrich Fraundorfer. Machine-learned regularization and polygonization of building segmentation masks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3098–3105. IEEE, 2021.