Multi-Class Segmentation from Aerial Views using Recursive Noise Diffusion

Benedikt Kolbeinsson
Imperial College London
bk915@imperial.ac.uk

Krystian Mikolajczyk
Imperial College London
k.mikolajczyk@imperial.ac.uk

Abstract

Semantic segmentation from aerial views is a crucial task for autonomous drones, as they rely on precise and accurate segmentation to navigate safely and efficiently. However, aerial images present unique challenges such as diverse viewpoints, extreme scale variations, and high scene complexity. In this paper, we propose an end-to-end multi-class semantic segmentation diffusion model that addresses these challenges. We introduce recursive denoising to allow information to propagate through the denoising process, as well as a hierarchical multi-scale approach that complements the diffusion process. Our method achieves promising results on the UAVid dataset and state-of-the-art performance on the Vaihingen Building segmentation benchmark. Being the first iteration of this method, it shows great promise for future improvements. Our code and models are available at: https://github.com/benediktkol/recursive-noise-diffusion

1. Introduction

Fully autonomous drones, also known as unmanned aerial vehicles (UAVs), have many socially and economically important applications, such as: infrastructure inspection, agriculture monitoring, search and rescue, disaster management, surveying and delivery services. However, drones need to understand their environment in order to perform these tasks autonomously [21]. Semantic segmentation is the task of labeling each pixel within an image to a class (e.g., “person” or “tree”) and is crucial for drones. Segmenting images from aerial views is especially challenging as they include diverse viewpoints, extreme scale variation and high scene complexity [32].

Conventional methods typically utilize a convolutional neural network (CNN) with an encoder-decoder structure, such as U-Net [42]. A limited receptive field will struggle with large scale variations, for example detecting large objects benefits from more global features while smaller objects are often better predicted using larger image sizes [49]. To overcome these limitations it is common to use multi-scale features [20, 36]. Combining scales by averaging the output from different scales at inference works but is improved when the scale combination is learned [10, 49].

To address this problem, we propose a hierarchical multi-scale diffusion model, which naturally learns how to combine multi-scale predictions (shown in Figure 1). This is made possible as we introduce a novel training method called recursive denoising, which allows a diffusion model to learn to use additional information during inference. We discuss the details of our method in Section 3 and demonstrate competitive results on UAVid [32] and state-of-the-art results on Vaihingen Buildings [16] in Section 4.

Our main contributions are:

- We introduce recursive denoising, which allows information to propagate through the denoising process, along with a hierarchical multi-scale approach.
- We implement a diffusion model for multi-class segmentation using recursive denoising.
- We achieve promising results on UAVid and state-of-the-art results on Vaihingen Buildings.
2. Related Work

Semantic Segmentation  Over the years, numerous methods have been proposed to tackle semantic segmentation, ranging from classical methods based on hand-crafted features to deep learning-based approaches. Long et al. [28] introduced fully convolutional neural networks (CNNs), which have subsequently emerged as the dominant paradigm for semantic segmentation [8, 9, 38]. The success of CNNs can be attributed to their ability to automatically learn meaningful representations from raw data, as well as their ability to capture context information through the use of large receptive fields [31]. In addition, many works have explored the use of different architectures and training strategies to improve the accuracy and efficiency of semantic segmentation models. For example, to increase the receptive field of a network, Fisher Yu and Vladlen Koltun [60] used dilated convolutions, later improved by Yu et al. [61] with dilated residual networks. PSPNet [63] and DeepLab [11] use pyramid pooling, while Tao et al. [49] and Chen et al. [10] use multi-scale attention to increase the network’s receptive field. Additionally, some approaches have used attention mechanisms [51] to selectively focus on informative regions of the image [48, 56, 57, 64].

For aerial views, Lyu et al. [32] propose a multi-scale-dilation network to combat the large scale variations. The use of attention is more common, where Wang et al. [52] develop a bilateral awareness network and Yi et al. [59] use a composite encoder from which multi-scale features are extracted. Yang et al. [58] use a dual branch approach, one branch for high resolution spacial details and the other for global aggregation. While, Wang et al. [53] explore using attention in the decoder stage for efficiency. A dual branch encoder with different scales and a cross-scale attention module is proposed by Ding et al. [19] with state-of-the-art results.

Overall, semantic segmentation remains an active and challenging research area with numerous opportunities for further exploration and improvement.

Diffusion  Denoising diffusion probabilistic models (DDPM), also known simply as diffusion models, are a type of generative models. Initially proposed by Sohl-Dickstein et al. [46], they have become state-of-the-art as they have been shown to outperform GANs [18]. Thus, they have recently become exceedingly popular for image generation [17, 40, 41, 43]. In addition, diffusion models are starting to be applied to a wide range of problems including object detection [12], video editing [35] and music generation [1].

Segmentation using generative methods  Generative adversarial networks (GANs) [22] have been used to generate synthetic data to train segmentation models [6, 50]. GANs are also used to predict semantic segmentation directly [30, 47, 62].

Baranchuk et al. [5] use intermediate representations from a pre-trained diffusion model to perform semantic segmentation. Amit et al. [2] train a diffusion model to perform segmentation but only for binary classification. Similarly, Wu et al. [55] use a diffusion model for binary segmentation for medical images. Both of these methods are inherently limited to binary classification. We propose modifications to diffusion models which allow us to perform full multi-class semantic segmentation. We show our method achieves state-of-the-art results on the Vaihingen building dataset and promising results on UAvid in Section 4.

Diffusion Theory  Diffusion models consist of two processes, first the diffusion process or forward process $q$, a Markov chain which gradually adds Gaussian noise to the data. Given a data distribution $x_0 \sim q(x_0)$, we define $q$ as:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \quad (1)$$

$$q(x_{t-1}|x_t) = \mathcal{N}(x_t; \mu_0(x_t), \Sigma_0(x_t, t)) \quad (2)$$

Where $\beta_t \in (0, 1)$ is the noise schedule, $x_1, \ldots, x_T$ are latent variables and $T$ is the number of time steps. The latent $x_T$ is approximately an isotropic Gaussian distribution, given a sufficiently large $T$.

The second process, called the denoising process or reverse process $p$, is also defined as a Markov chain starting at $p_0(x_T) = \mathcal{N}(x_T; 0, I)$ but learns to gradually remove Gaussian noise:

$$p_0(x_{0:T}) = p_0(x_T) \prod_{t=1}^{T} p_0(x_{t-1}|x_t) \quad (3)$$

$$p_0(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_0(x_t, t), \Sigma_0(x_t, t)) \quad (4)$$

Interestingly, given $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$, sampling $x_t$ at arbitrary timestep $t$ is achieved with:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \quad (5)$$

Further details on the theory behind diffusion models may be found in [25, 37]. However, Bansal et al. [4] call this theory into question. They show diffusion models can be trained without Gaussian noise and even with deterministic image degradation. We are inspired by their work, and thus we propose to use a diffusion model to predict semantic segmentation maps. Converting diffusion models from typical generative tasks, to a predictive task, involves modifying the diffusion process and the denoising process. We discuss the details in Section 3.
3. Recursive Noise Diffusion

Given an aerial RGB image \( x \in \mathbb{R}^{W \times H \times 3} \), our goal is to predict a semantic segmentation map \( s \in \mathbb{R}^{W \times H \times d_{\text{classes}}} \), with corresponding class labels for each pixel. Where \( W, H \) are the width and height of the image, and \( d_{\text{classes}} \) is the total number of classes.

### 3.1. Multi-class diffusion

We modify the diffusion process to better suit the problem of predicting semantic segmentation. Given an image \( x \in \mathbb{X} \), the corresponding one-hot encoded segmentation map \( s_0 \in (S|x) \) and time step (noise level) \( t \in [0, T) \), we define the forward noising process \( q \), which adds Gaussian noise with variance \( \beta_t \in (0, 1) \), as follows:

\[
q(s_t | s_{t-1}) = s_{t-1} + \mathcal{N}(0, \beta_t I)
\]

We define the noise schedule \( \beta_t \), as:

\[
\beta_t = \frac{t}{T}
\]

Trivially, the total added noise can be written as:

\[
\epsilon_t = s_t - s_0
\]

We approximate the reverse process (denoising) using a neural network, \( \epsilon_\theta \). Following Ho et al. [25], \( \epsilon_\theta \) predicts \( \epsilon_t \), meaning:

\[
\epsilon_t \approx \epsilon_\theta(s_t, x, t)
\]

Thus, we can predict the segmentation map at any arbitrary time step \( t \), as follows:

\[
s_0 \approx s_t - \epsilon_\theta(s_t, x, t)
\]

For training, we use the mean squared error (MSE) of the predicted noise as our loss function:

\[
L_{\text{mse}} = E_{s_0, x, t, \epsilon_t} \left[ ||\epsilon_t - \epsilon_\theta(s_t, x, t)||^2 \right]
\]

### 3.2. Recursive denoising

When we train a model on an arbitrary step of the noising process, as is most common [25, 37], we notice the model quickly overfits on the training data and does not generalize. We believe the binary labels allow for a trivial denoising strategy to be learned, e.g. rounding logits. This causes two issues, first, the model does not fully utilize the conditional image but rather simply uses the noisy segmentation. Second, during testing, the model is too dependent on the initial steps in the denoising process. To solve this, we propose to train with recursive denoising.

Training with recursive denoising involves progressing through each time step \( t \) from \( T \) to 1, recursively (as the name suggests), which allows a portion of the predicted error to propagate. Figure 2 shows an overview of the recursive noise diffusion method. The first step of the approach involves the use of pure noise. The subsequent stage utilizes a noise function to diffuse the previous predicted segmentation. Following this, a diffusion model is employed to denoise the diffused segmentation, leveraging a conditioning RGB image. Finally, the denoised predicted segmentation is compared to the ground truth. Importantly, it is worth noting that the ground truth segmentation is never utilized as a component of the input to the model. Algorithm 1 outlines the steps involved in processing a single training sample. Our recursive denoising approach can be conceptualized as emulating the testing process, which necessitates advancing through each time step.

An essential characteristic of our proposed method is its agnosticism to the selection of the noise function, diffusion model, and loss function. This feature enables the flexibility to choose the most suitable components for a specific task or dataset. To illustrate this capability, we demonstrate the use of a distinct noise function, diffusion model, and loss function in Section 4.1 compared to Section 4.2 of our experiments. Such flexibility provides greater versatility and adaptability to the proposed approach, enabling it to
be widely applicable in a range of tasks and settings.

Recursive denoising can also serve as a means of transmitting supplementary information, as elaborated upon in Section 3.3.

### Algorithm 1: Training with recursive denoising

Input: \( x \in \mathbb{R}^{W \times H \times 3} \), RGB image  

Input: \( \bar{s} \in \mathbb{R}^{W \times H \times \text{classes}} \), segmentation labels

Parameters: \( T \in \mathbb{Z}^{+} \), number of time steps

1. \( \hat{s}_T \sim \mathcal{N}(0, I) \)
2. for \( t = T, \ldots, 1 \) do
3. \( z_t \sim \mathcal{N}(0, I) \)
4. \( s'_t \leftarrow \hat{s}_t + z_t \times \frac{1}{T} \) // diffuse
5. \( \hat{s}_{t-1} \leftarrow s'_t - \epsilon_t(s'_t, x, t) \) // denoise
6. \( l \leftarrow \| \epsilon_t(s'_t, x, t) - (s'_t - \hat{s}) \|^2 \)
7. Update \( \epsilon_t \) w.r.t. \( l \)
8. end

### 3.3. Hierarchical multi-scale diffusion

Performing segmentation at multiple image scales can improve the prediction [19]. We can exploit the propagation artifact from recursive denoising using a hierarchical multi-scale denoising process, shown in Figure 3. The assumption underlying our approach is that the model can extract valuable information from the previous noisy segmentation to improve the accuracy of the next prediction. Leveraging our recursive denoising approach, this assumption can be exploited by enabling the model to make predictions on a smaller (scaled down) image first, thereby capturing larger objects and contextual information, followed by predictions on a higher resolution image to capture finer details and smaller objects.

We define a simple linear scaling schedule to determine when to change scales. Our scaling schedule evenly divides the time steps for each scale, starting with the smallest scale. Our approach is not restricted solely to this particular scaling schedule. In Section 4.2, we investigate a variation of this scaling approach.

### 4. Experiments

We demonstrate the versatility of recursive diffusion in this section through two experiments: multi-class segmentation and binary segmentation. Each experiment uses a distinct combination of model, noise function, and loss. Experimental settings and results for each experiment are presented separately.

#### 4.1. Multi-class segmentation

In this section, we demonstrate the effectiveness of our method for multi-class segmentation for aerial views. Multi-class segmentation refers to the process of labeling each pixel in an image into more than two distinct classes. This is in contrast to binary segmentation, which involves the separation of pixels in an image into two classes only. Multi-class segmentation is generally more challenging than binary segmentation, as it requires the accurate identification and labeling of multiple objects or regions within an image, each belonging to a distinct class.

**Data** We use UA Vid [32], a specialized drone dataset, containing a total of 420 (4K resolution) images from aerial views. The data is split into sets of 200, 70, and 150 images for training, validation and testing, respectively. There are eight classes; Building, Road, Tree, Low Vegetation, Moving Car, Static Car, Human and Clutter.

**Metrics** We report the Intersection over Union (IoU), also known as the Jaccard Index, for individual classes as well as the mean Intersection over Union (mIoU), averaged over all classes. These metrics are often expressed as percentages.

\[
\text{IoU} = \frac{TP}{TP + FP + FN} \quad (12)
\]

\[
\text{mIoU} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i + FN_i} \quad (13)
\]

Where \( TP, FP \) and \( FN \) are the true positive, false positive and false negative between the predicted and ground truth class labels, respectively. \( N \) is the total number of classes in the dataset. We also report the F1-score.

**Architecture** As the recursive denoising approach is independent of the choice of model, it provides us with the flexibility to choose an efficient and effective model for multi-class segmentation of aerial views, such as UNetFormer [53]. UNetFormer can be conveniently transformed into a diffusion model by incorporating the prior segmentation as an extra input, which requires only minor modifications to the original architecture. The architectural changes are illustrated in Figure 4. Specifically, the previous segmentation input is scaled (using bilinear interpolation) and concatenated to minimize the introduction of additional network complexity. The resulting WNetFormer, which is a variant of UNetFormer with these modifications, has only a marginal increase in parameters compared to the original UNetFormer, with 11.8 M parameters versus 11.7 M parameters.

**Loss** To demonstrate the versatility of our approach, we employ a loss function similar to that of UNetFormer. UNetFormer incorporates both a primary loss and an auxiliary loss, with the latter being applied to an auxiliary head that...
integrates features from the global-local transformer block (GLTB). Specifically, the auxiliary loss takes the form of a cross entropy loss. It is noteworthy that we have retained this aspect of the UNetFormer methodology and have not made any modifications to the auxiliary loss function. However, we simplify UNetFormer’s primary loss as we only use the cross entropy portion. In addition, we target the added noise instead of directly predicting the segmentation, as discussed in Section 3.

**Noise function**  As previously mentioned, recursive noise diffusion does not impose any restrictions on the noise function. However, care must be taken when choosing a noise function as the one-hot encoded binary labels for the classes is inherently robust to low levels of noise. We propose a Softmax-Average noise function which is both effective and controllable. Using the same notation as in Section 3, the Softmax-Average noise function is defined per pixel as:

$$s'_{t,s} = softmax ((1 - \lambda_{noise,t}) \times \hat{s} + \lambda_{noise,t} \times z) \tag{14}$$

Where $z$ is a softmaxed vector of random noise where each element is sampled from the normal distribution, and

$$\lambda_{noise,t} = \frac{T - t}{T} \tag{15}$$

**Training**  Diffusion models tend to use a large number of time steps, in the range of hundreds or thousands [37]. However, we find we can train with far fewer time steps (as low as 2), which directly translates to reduced inference time. However, performance is better when trained with more time steps as can be seen in Table 2, and we train with
Table 1. Comparison of different methods on the UA Vid [32] test dataset. Each column represents IoU per respective class with the right-most column being the mean IoU.

Table 2. The impact of varying the number of time steps during training, on UA Vid validation. More time steps correspond to higher mIoU. Best results are highlighted in bold.

Table 3. The impact of varying inference time steps. The effects of different time steps during inference using a model trained with 128 time steps. This approach allows for efficiency improvements while maintaining high performance. Best results are highlighted in bold.

Table 4. The impact of our hierarchical multi-scale approach. We present performance results in terms of relative mIoU improvement compared to our recursive method without multiple scales on the UA Vid dataset. We progressively introduce additional scales and not excessively small scales contain limited additional information. Best results are highlighted in bold.

Results Our proposed recursive denoising method achieves promising results, shown in Table 1. The method outperforms other recently proposed approaches including BANet [52] and UAVFormer [59]. Qualitative experiments are shown in Appendix A. As this is the first iteration of this method there is scope for improvement through optimization of hyperparameters that were not fully explored due to limited computational resources. The Human class is particularly challenging, as it is both underrepresented and each object is small and isolated in relation of the entire scene. We investigate the effects of varying the number of time steps during training. As presented in Table 2, a discernible trend emerges, indicating that higher numbers of training time steps yield higher segmentation scores.

In order to enhance the efficiency of our method, we further examine the impact of skipping time steps during inference, as illustrated in Table 3. This is made feasible through the utilization of Equation (10). Our findings reveal that a reduction of up to 50% in the number of time steps can be implemented with minimal effect on performance. However, beyond this threshold, a sharp decline in performance is observed. Therefore, our results demonstrate that our method can be trained with a high number of time steps while simultaneously enhancing its efficiency during inference, without any significant loss of performance.

In addition, we verify the effects of our hierarchical multi-scale approach and demonstrate its direct benefits when used with recursive denoising in an ablation study shown in Table 4.
4.2. Binary segmentation

The present study provides evidence supporting the effectiveness of our method in binary segmentation settings. Specifically, our results demonstrate that our method can successfully achieve state-of-the-art results in this context.

Data We experiment on Vaihingen Buildings [16], specifically the setup used in [33], which contains 168 images. These are split between a training set of 100 images and a test set of 68 images. The objective is to label each pixel as either the central building or as background. Multiple buildings can be present in a single image, however, only the building in the center of the image should be labeled. We use the same augmentation strategy as for UAVid but in addition, we also add random (50%) vertical flips.

Noise function and loss We find a simple noise function works best for the binary segmentation setting. Using the same notation as in Section 3, the noise function is defined per pixel as:

\[ s_{t,s}' = \hat{s}_t + z_t \times \frac{t}{T}, \]

Where \( z_t \) is sampled from a normal distribution.

For the loss function we use MSE of the predicted added noise as shown in Equation (11).

Architecture Our model consists of four modules, a time step head, an image encoder head, a diffused segmentation encoder head and finally the main UNet-like encoder-decoder. The time step head simply converts the time step to a sinusoidal positional embedding, similar to the positional embeddings of [51]. The image and segmentation heads share a similar structure, consisting of two ResNetBlocks. The output is added together, similar to [2]. We take inspiration from Efficient U-Net [43] to design our encoder-decoder module. It combines the time step embeddings at multiple stages through the network (in each ResNetBlock). It also utilizes Efficient Attention [45] which is a type of attention with linear complexity. The output of the model is the predicted error (or noise) of the noisy segmentation. The model configuration is shown in Appendix B.

Training For training, we augment the data with random (50%) horizontal and vertical flips. In addition, random augmentations to the contrast, saturation and hue. We use AdamW [29] as our optimizer with an initial learning rate of \( 5 \times 10^{-5} \), decay gamma of 0.95 and weight decay of \( 1 \times 10^{-3} \). We use gradient clipping at 1.0 and train for 70 epochs.

We make a modification to the multi-scale schedule for this task. We convert the linear schedule to a repetitive schedule, which means at each time step, the input is scaled to a different size. This process is shown in Appendix C.

Results The qualitative results of our method on Vaihingen Buildings are presented in Figure 5, illustrating its abil-
ity to produce segmentations with sharp edges, albeit with some instances of slightly rounded corners. Moreover, Table 5 provides quantitative results on Vaihingen Buildings, where our method achieves the best mIoU.

An analysis of the performance of different model types reveals that diffusion-based models, such as our proposed method and SegDiff, outperform more conventional segmentation approaches. Not only does our method outperform SegDiff but it also supports multi-class segmentation.

5. Discussion

Our experiments have demonstrated the feasibility of utilizing diffusion models for multi-class segmentation tasks, yielding results that are comparable to conventional methods while also indicating a higher potential for diffusion models. One possible explanation for this success is the recursive diffusion process, which mimics the training process, and enables the inherent learning of any biases in the noise reduction process. This is the first iteration of our proposed method, and our results suggest that there is ample potential for further research in this area.

Furthermore, it is worth noting that the results presented in this study were obtained with a limited amount of computational resources. The use of more powerful computational resources could potentially yield even better results by enabling the utilization of larger models and longer training times. This is supported by trend seen in Table 2. Additionally, more complex diffusion models and variations thereof could be explored, potentially leading to even greater performance gains. Therefore, further investigation into the use of diffusion models for multi-class segmentation tasks could yield even more advances.

Despite the promising results obtained from utilizing diffusion models for multi-class segmentation, it is important to acknowledge the limitations of this approach. One notable limitation is the higher inference time required for diffusion models, which manifests as a trade-off between accuracy and computational efficiency, as shown in Table 3. Depending on the specific application, the appropriate balance between accuracy and inference time needs to be considered. However, this limitation is being partly mitigated with the ongoing advancements in hardware capabilities and knowledge distillation techniques [27, 34]. These developments enable the execution of highly accurate models on smaller and more cost-effective devices. Therefore, the use of diffusion models for multi-class segmentation tasks remains a promising option.

Another limitation is the necessity for labeled training data. However, this limitation is a common feature of all supervised segmentation methods. Acquiring sufficient and accurate labeled training data can be a challenging and time-consuming task, especially when dealing with more complex segmentation tasks. The integration of synthetic data into supervised methods for computer vision has gained momentum as a promising solution to address the challenges posed by limited real-world annotated data [3, 13]. By augmenting training datasets with synthetically generated samples, these methods aim to enhance generalization and robustness, enabling improved performance across various tasks [43].

It is interesting to note, the current state-of-the-art for image generation uses diffusion models. Here we use a similar concept for a different purpose. Therefore it would be interesting to explore potential synergies.

The use of more powerful hardware can provide the necessary resources to train and run larger and more complex diffusion models, which can potentially lead to further performance gains [7, 26, 39]. Additionally, future work can also focus on incorporating recent algorithms and techniques designed to reduce the computational overhead associated with diffusion models [23, 54], thereby improving their practicality in real-world applications.

While the utilization of diffusion models for aerial image segmentation tasks has shown promising results, it is worth exploring the applicability of this approach to other segmentation tasks, such as urban street scene segmentation. One such dataset that is commonly used for this task is the Cityscapes [15] dataset, which consists of street scenes from various urban environments. Applying diffusion models to this dataset can potentially provide significant improvements in the accuracy of street scene segmentation, enabling the detection of more fine-grained details and features. However, it is important to consider specific factors that can affect the performance of diffusion models in this different setting. Nonetheless, the potential benefits of utilizing diffusion models for urban street scene segmentation make it an interesting area for further investigation.

Furthermore, a key area for future investigation involves assessing how semantic segmentation diffusion models generalize across different datasets. Analyzing their performance when trained on one dataset and applied to another could reveal insights into their adaptability and robustness.

6. Conclusion

In this work, we present recursive denoising along with a hierarchical multi-scale diffusion model for semantic segmentation from aerial views. Recursive denoising allows for information to propagate through the denoising process. We show our proposed solution yields promising results on UAVid and state-of-the-art results on Vaihingen Buildings. We believe our recursive denoising diffusion model is only the first step of a new promising class of segmentation models. Improving aerial segmentation can unlock or improve real-world applications such as delivery, maintenance and, search and rescue, leading to increased efficiency, reduced costs, and improved safety across industries.
References


[52] Libo Wang, Rui Li, Dongzhi Wang, Chenxi Duan, Teng Wang, and Xiaoliang Meng. Transformer meets convolution:


