Modernized Training of U-Net for Aerial Semantic Segmentation

Jakub Straka
Department of Cybernetics
Technická 8, 301 00 Plzeň, Czech Republic
strakajk@kky.zcu.cz

Ivan Gruber
Department of Cybernetics and New Technologies for the Information Society
Technická 8, 301 00 Plzeň, Czech Republic
grubiv@ntis.zcu.cz

Abstract

In this paper, we propose an improved training protocol of U-Net architecture for the semantic segmentation of aerial images. We test our approach on the challenging FLAIR #2 dataset. We present an extensive ablation study on the influence of different approach components on the overall performance. The ablation study includes a comparison of different model backbones, image augmentations, learning rate schedulers, loss functions, and training procedures. We additionally propose a two-stage training procedure and evaluate different options for the model ensemble. Based on the results we design the final setup of the model training protocol. This final setup decreases the relative error by approximately 18% and achieves mIoU equal to 0.641, which is a new state-of-the-art result. Our code is available at: https://github.com/strakaj/U-Net-for-remote-sensing.

1. Introduction

The rapidly growing human population affects the ecosystem and is dependent on efficient agriculture and urban development and planning. To support these developments and with the new possibilities that modern technologies offer, the data domain of aerial and satellite images become more and more popular. The utilization of these data finds application in many areas of modern society, the non-exhaustive list of examples follows: Land-cover mapping and classification [20] whose main goal is to determine the use and distribution of individual surface features that can help with monitoring and development of the natural environments. Natural disaster detection [12, 17, 18] as, floods, which very often happen in specific parts of the world, and claim many victims. The use of aerial images can help with faster detection of disasters, help with assessing damage, and help with better management of the environment and infrastructure. Plant monitoring and disease detection [16, 19] is an integral part of modern agriculture. Diseases can destroy large areas of crops, resulting in food scarcity in certain global regions. High-altitude images can help monitor plant health and better target pesticide and fertilizer use.

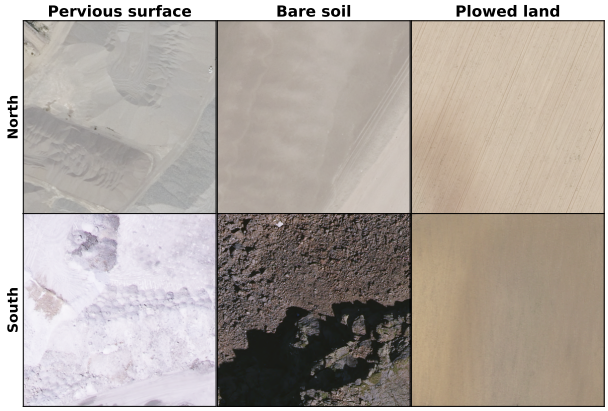

Figure 1. Comparison of patches of the same class taken in different parts of the country.

Semantic segmentation of larger areas such as the whole country brings several challenges. Most notably inter-class similarity and intra-class dissimilarity. Some classes, especially classes that distinguish different types of crops and trees, can be difficult to recognize from a height-altitude, which can lead to confusion. Representatives of one class can vary drastically depending on different geographical and climatic conditions, which makes classifications difficult. An example of such a phenomenon is shown in Figure 1. Another factor that can alter the appearance of the landscape is the weather and the season. All these factors

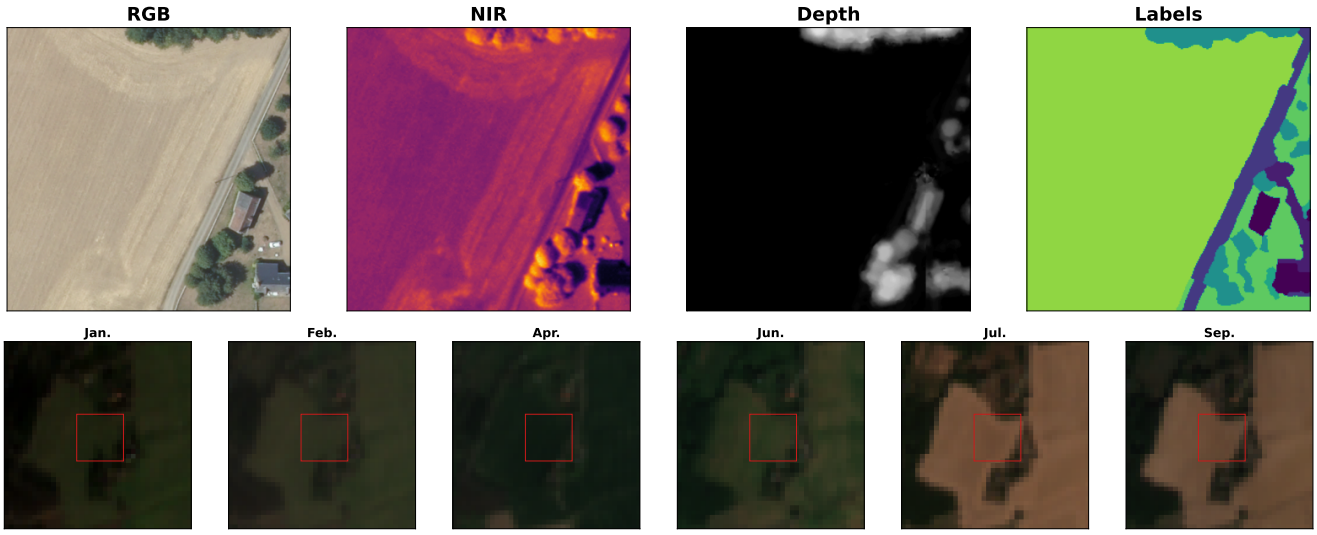

Figure 2. Sample input data. The first line shows the aerial data and the corresponding labels. The second row shows satellite images from several months. The red frame marks the area that corresponds to the aerial image.

make semantic segmentation of aerial images a challenging and complex task.

The French National Institute of Geographical and Forest Information (IGN) [1] issued FLAIR #1 [8] and FLAIR #2 [7] challenges with the intent to monitor land-cover and create high-resolution land-cover maps. The goal of both challenges was semantic segmentation of the land cover of France. FLAIR #1 was mainly focused on the use of high-resolution data, whereas, FLAIR #2 added satellite images from Sentinel-2 [6] and presented the challenge of fusion of high-resolution aerial data with low-resolution satellite data.

In this work, we propose an improved training protocol for the U-Net-based model [22] for aerial and satellite data segmentation. The used model is based on a baseline model proposed within the FLAIR #2 challenge. We conducted an extensive ablation study on hyperparameters, and different model backbones, and introduced a two-staged training procedure. Based on the results of the ablation study we proposed the final model ensemble, which improved the mean Intersection-over-Union (mIoU) from the original 0.576 to **0.641** and achieved new state-of-the-art results on the FLAIR #2 dataset.

2. Related Work

2.1. Aerial and satellite datasets

There are many applications for satellite and aerial images, which results in a large variety of available datasets. The datasets can differ in geographical location (urban vs. rural areas), but also in in the resolution of the data (satellite vs. aerial). High-resolution images are often obtained using aerial surveys and more often focus on smaller urban areas. The INRIA dataset [15] focuses on the segmentation of buildings in 5 dissimilar cities from RGB images with spatial resolution 0.3m. Similarly, the ISPRS dataset[1] covers 2 cities with a spatial resolution of 0.09m and 0.05m. High resolution of data allows for the segmentation of classes such as *car* or *tree*. In contrast, datasets such as LandCoverNet [3] based on the Sentinel-2 [2] data with spatial resolution 10m or the GID [25] dataset based on satellite images from Gaofen-2 with spatial resolution 4m contain macro-level classes such as *cultivated vegetation*, *water*, *bare ground*, or *farmland*. The LandCoverNet dataset is taken mainly over the rural areas from different parts of the world. The GID dataset covers urban and rural areas in China. The LoveDA [26] dataset with spatial resolution 0.3m focuses more on covering both urban and rural areas as each has a different class distribution.

2.2. Aerial and satellite methods

One of the most common models used for segmentation is U-Net [22]. In the work [4], authors used U-Net with reduced number of convolutional kernels for deforestation monitoring. U-Net was used for the segmentation of satellite images of one area over several years. Based on a comparison of the segmentation masks, the area of the forest was assessed. Similarly, authors in the [27] used U-Net for rice field segmentation and field boundary detection from satellite images. In study [28] authors compared a variety of segmentation models for wildfire segmentation from aerial images. Two models that achieved the best results

[1] https://www.isprs.org/education/benchmarks/UrbanSemLab

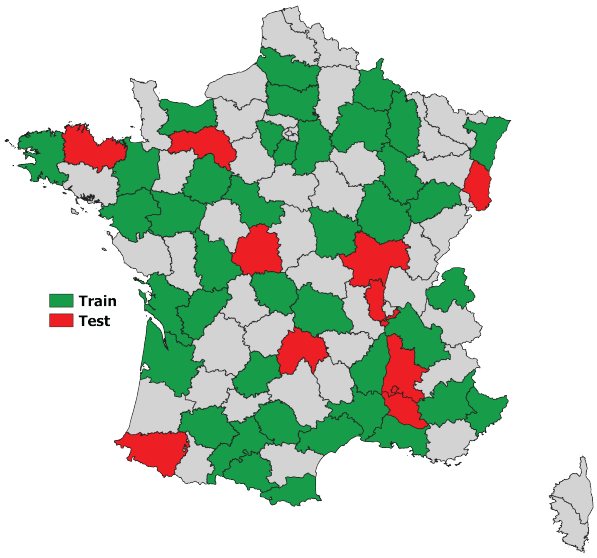

Figure 3. France sub-regions used in the FLAIR #2 dataset. Image source: [7]

were U-Net and DeepLabV3 [5]. In work [12] the authors addressed the problem of common floods by constructing a system for real-time flood segmentation from aerial images. PSPNet [31], DeepLabV3, and U-Net were tested as the segmentation method within the proposed system. All methods were compared on the FloodNet [21] dataset where the best results were achieved with PSPNet.

3. Data

The FLAIR #2 challenge dataset contains data from 50 sub-regions of France. Regions used in the dataset are shown in Figure 3. Sub-regions are divided into patches that have the same size across the dataset. The number of patches is dependent on the sub-region size. The dataset contains aerial and satellite images. The aerial images were annotated with 18 land cover classes and *other class* that correspond to pixels that could not be assigned to any of the 18 classes. Because some classes are represented sparsely in the dataset, 18 land cover classes were reduced to 12 by grouping 6 classes with the lowest representation with *other class*.

The aerial images were taken by plane and have high resolution. Each patch has a resolution of 512×512 pixels. The spatial resolution of one pixel is 0.2 m. In addition to visible RGB spectral bands, each patch also contains an image in a near-infra-red (NIR) spectral band and a depth map. Each patch is annotated with a segmentation mask and metadata which contain the position of the patch, date, and camera that was used to capture the image. The dataset contains a total of 61712 annotated images in the training set and 16050 images in the test set.

The satellite images were captured by Sentinel-2 and have a much lower resolution. The resolution of each image is 40×40 pixels and the spatial resolution of one pixel is 10 m. Compared to the aerial images the satellite images cover a broader spectral range, consisting of a total of 10 spectral bands spanning from the visible to the medium infrared spectrum. Satellite images should serve as a support for aerial images and should provide spatial context. Therefore, the satellite images are centered at aerial image patches and only 10×10 center pixels correspond to the aerial patch. Another difference from the aerial images is that the satellite images contain multiple images of the same area. The images were taken over several years. This helped to capture changes in the area of the patch in different seasons. The satellite images are not annotated.

Additionally, the FLAIR #2 [7] challenge has the two following constraints: 1) Use of external data is prohibited, 2) Inference time of the proposed method should not exceed 2.5 times the inference time of the baseline method. Our approach is designed in a way to hold these constraints.

4. Baseline method

The baseline model **U-T&T** (Textural and Temporal information) presented in [7] is composed of two branches. The first and main branch is the standard U-Net. This branch is used to process aerial images. ResNet-34 [11] pre-trained on the ImageNet [23] dataset is used as a backbone. This branch processes RGB aerial images concatenated with a corresponding NIR image and a depth map.

The second branch is the U-TAE [10] model. This branch processes satellite images and serves as a support for the first branch. The model is also based on U-Net with modifications that take into account the temporal nature of the satellite data. The variable temporal dimension is addressed by the Temporal Attention Encoder (TAE) proposed in [9]. The output of the second branch is fused with the feature maps on all feature levels of the main branch encoder.

The fusion module proposed in [7] is decomposed into *cropped* and *collapsed* sub-modules. Input into the fusion module is the output of the U-TAE branch. The *cropped* sub-module aims to incorporate information embedded into the satellite image that corresponds to the area of the aerial image. The *collapsed* sub-module preserves spatial information from the whole satellite image. Outputs of both sub-modules are added pixel-wise together with the U-Net feature map.

Each branch of the model is supervised separately by cross-entropy loss. The final loss is obtained as the sum of these two losses. The baseline method uses the SGD optimizer. The best baseline model uses additional procedures to achieve better results. Satellite images that contained clouds are excluded and satellite images for each month are averaged to reduce the number of input images. During the

training, geometric augmentations such as horizontal and vertical flip together with rotation by 0, 90, 180, or 270 degrees are included. Additional procedures are described such as randomly dropping U-TAE modality and the usage of metadata, however, neither of them improved overall performance.

5. Experiments

In this section, we will describe all experiments we conducted within our ablation study. We focus on improving the baseline method from Section 4 with a goal to not only reach the best results possible but also to compare the benefits of various hyperparameters to the overall performance of the model.

Most of the results are reported as a mean of three runs. Each run was initialized with a different seed. Values after $\pm$ are the standard deviation of these three runs. In some experiments, we conducted preliminary experiments to narrow down the range of possibilities. In these instances, the experiment was run only once, and as a result, standard deviation is not reported.

The FLAIR #2 dataset does not provide a fixed training and validation set. The validation set is therefore randomly selected as 20% of all regions. To make the comparison fair, we fixed the training and validation set so that it does not depend on the seed. This resulted in 47712 images in the training set and 14000 images in the validation set. Each aerial image is loaded as an image with five channels, three channels correspond to RGB spectral bands, one to NIR band, and one to depth data.

We started the experiments with the following settings, which we gradually modified based on the results. Same as in the challenge paper [7], we filtered satellite images with clouds, averaged the satellite images from the same place but different months, used the U-TAE branch, omitted the usage of metadata, and used the same augmentations as in the original paper. All models were initially trained with a constant learning rate of 0.001, cross-entropy loss, batch size 10, 12 epochs, and the AdamW optimizer [14]. All models of the main branch were pre-trained on ImageNet, whereas, the U-TAE branch was randomly initialized. If not stated otherwise, the experiments are evaluated on the validation set mentioned above.

Backbone. We first evaluated different backbone models for the main model branch, specifically ResNet [11], ResNeXt [30], and MiT [29]. The first two models are convolutional-based, and the last one is a transformer-based model. Due to the ImageNet pretraining, all models expected the input to have 3 channels, but our input data had 5 channels (RGB+NIR+depth). We solved this issue by modification of the first layer in the case of convolution-based models and by addition of 1 convolutional layer with kernel size 1×1 at the start of the transformer-based model which reduced the number of channels from 5 to 3. For each model, we tested its smaller and larger variant.

In Table 1 are results from the initial experiments. Unexpectedly, the smaller variants of models achieved better results than the larger ones except for ResNeXt. Due to the inferior performance of the larger models and inference time constraints, we limited the further experiments to the smaller variants. In Table 2 are the results of these models averaged over the three runs. In the following experiments, we continued to use ResNet-34 due to the best trade-off between performance and training speed.

Backbone	Params (M)	mIoU
ResNet-34	**21**	**0.566**
ResNet-50	23	0.532
ResNeXt-50-32x4d	22	0.551
ResNeXt-101-32x4d	**42**	**0.585**
MiT-B2	**24**	**0.584**
MiT-B3	44	0.543

Table 1. Initial comparison of backbones.

Backbone	mIoU
ResNet-34	0.557 $\pm$ 0.030
ResNeXt-50-32x4d	0.559 $\pm$ 0.016
MiT-B2	**0.566** $\pm$ **0.033**

Table 2. Detailed comparison of the smaller backbones.

Augmentations. The baseline method used horizontal and vertical flip and rotation as augmentations. We attempted to improve the model's robustness by introducing additional augmentations. We chose shift/scale/rotate augmentation with a shift limit of 0.2, scale limit of 0.15, and rotate limit of 20. A border extrapolation method was set to reflect.

Furthermore, we added a coarse dropout augmentation that randomly removes parts of the image. Number of removed areas was in the range $\langle 2, 8 \rangle$. The size of the removed areas was set in the range $\langle 16 - 48, 16 - 48 \rangle$. The aim was to encourage the model to rely on the surrounding context to infer the pixel class.

Unfortunately, from results in Table 3, it can be seen that neither of the augmentations improves the performance. We argue this is caused by the relatively small dataset size which results in the ineffectiveness of more complex augmentation methods. Moreover, it is a well-known fact that bigger models benefited from augmentations more than the smaller ones due to their higher capacity. Based on these results continued using only the three original augmentations.

flip-h	flip-v	rot90	shift scale rotate	drop	mIoU
					0.547± 0.009
✓	✓	✓			**0.557± 0.030**
✓	✓	✓	✓		0.551± 0.028
✓	✓	✓		✓	0.545± 0.035
✓	✓	✓	✓	✓	0.553± 0.014

Table 3. Comparison of effect of augmentations.

Scheduler. Learning rate is one of the most important hyperparameters during training. We conducted experiments on multiple learning rate schedulers with different parameters and starting values of the learning rate. To shorten the training time, the initial experiments, reported in Table 4, were performed only with half of the training data, and the number of epochs was decreased to 6. The validation set stayed the same.

In Table 5 are the results of the best three schedulers and constant learning rate, which was our baseline. The initial learning rate was set to 0.0001 for all schedulers. The number of epochs is also 6 this time, but we trained on the whole training set. The best performance is achieved by the model with a multi-step scheduler which dropped by 90% of the learning at each drop step. Based on the results, we continued in the following experiments with an initial learning rate of 0.0001 and a multi-step scheduler. During additional testings, we also observed that additional epochs after the 6th epoch did not provide any benefits to the final performance, therefore, we continue with only 6 training epochs in the following experiments.

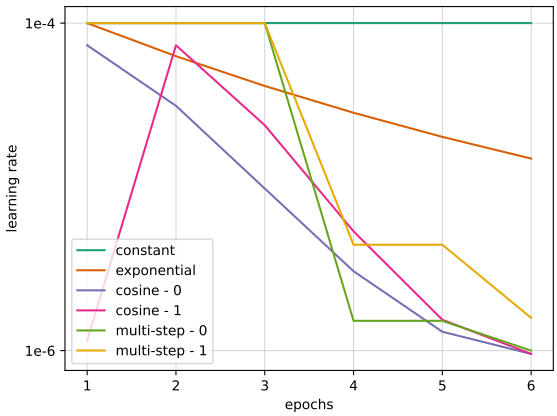

Figure 4. Tested learning rate schedulers.

Loss. Another crucial component of the training procedure is the loss function. Except for commonly used cross-entropy loss, there are many other widely used loss functions in semantic segmentation such as dice loss [24], and focal loss [13]. Another common practice is label smoothing which is a regularization technique that sets the target value for the correct class in a one-hot vector to (1-ϵ) and the values for incorrect classes to $\frac{\epsilon}{c-1}$, where c is the number of classes and ϵ is a smoothing factor.

In our experiments, if applicable, we used $\epsilon = 0.2$. We used a α-balanced version of the focal loss as described in [13]. Parameter γ was set to 2 for all experiments with focal loss. Due to the fact that experiments in this subsection were run parallel with experiments from the previous subsection (schedulers), we trained models with a constant learning rate and starting learning rate equal to 0.0001. In Table 6 are the results for different loss functions. The best

Scheduler	Learning rate	mIoU
constant (baseline)	0.001	0.515
constant	0.0001	0.544
constant	0.00001	0.530
exponential	0.001	0.439
exponential	0.0001	0.552
exponential	0.00001	0.514
cosine - 0	0.001	0.487
cosine - 0	**0.0001**	**0.573**
cosine - 0	0.00001	0.423
cosine - 1	0.001	0.532
cosine - 1	0.0001	0.561
cosine - 1	0.00001	0.412
multi-step - 0	0.001	0.527
multi step - 0	**0.0001**	**0.575**
multi-step - 0	0.00001	0.449
multi-step - 1	0.001	0.542
multi-step - 1	0.0001	0.556
multi-step - 1	0.00001	0.512

Table 4. Initial scheduler experiment. Training data were limited to half and the number of epochs to 6. The reported results are on the full validation set.

Scheduler	mIoU
constant	0.577±0.009
cosine - 0	0.586±0.007
cosine - 1	0.578±0.018
multi-step - 0	**0.589±0.006**

Table 5. Detailed comparison of the best schedulers. trained on the whole training set.

results achieved focal loss with label smoothing[2]. In the final setup, we therefore used focal loss with label smoothing.

Loss	α	mIoU
cross-entropy	-	0.577±0.009
cross-entropy - smooth	-	0.580±0.002
dice	-	0.581±0.011
focal	1	0.556±0.005
focal	0.25	0.594±0.016
focal - smooth	1	**0.607±0.007**

Table 6. Comparison of loss functions. For all experiments with focal loss was γ set to 2 and for experiments with label smoothing was ϵ set to 0.2

U-Net pre-training. The main part of the model is U-Net which processes aerial images. The second part is U-TAE which supports the main network by adding information from satellite images. We wanted to emphasize the importance of the main network by the procedure where we first train the U-Net only and then finetune the full model with the U-Net weights initialized from the pre-trained U-Net and the U-TAE branch randomly initialized.

In the first part of Table 7, there are the results of U-Net trained for 6 epochs. In the second part, there are the results of full models trained from the U-Net weights trained in the first part. The full model was trained by an additional 6 epochs. In the last part of the table, there are the results of the full model trained without the proposed U-Net pre-training. It should be noted that except for ResNeXt, all the models benefited from the two-stage training procedure. Based on the results we decided to employ this strategy in the final setup.

Ensemble. A model ensemble is a common technique used to combine the predictions of multiple models to improve overall performance. We verified that this technique helps also in our case. We trained three models with the same backbone but each with a different seed. Models were trained in two stages with focal loss and a multi-step learning rate scheduler. In Table 8 are the results of the three models evaluated on validation and test set and test results of the model ensemble created from these three models. Model ensemble achieved significantly better results than individual models which confirmed the benefits of the model ensemble.

[2]Implementation: https://github.com/Kageshimasu/focal-loss-with-smoothing

Model	Backbone	Pre-trained U-Net	mIoU
U-Net	ResNet-34		0.576±0.010
U-Net	ResNeXt-50-32x4d		0.563±0.011
U-Net	Mit-B2		0.602±0.010
U-T&T	ResNet-34	✓	0.616±0.002
U-T&T	ResNeXt-50-32x4d	✓	0.605±0.007
U-T&T	**Mit-B2**	✓	**0.626±0.012**
U-T&T	ResNet-34		0.597±0.005
U-T&T	ResNeXt-50-32x4d		0.611±0.013
U-T&T	Mit-B2		0.608±0.003

Table 7. Comparison of models initialized with pre-trained U-Net.

Model	Backbone	Validation mIoU	Test mIoU
U-T&T	MiT-B2	0.632	0.604
U-T&T	MiT-B2	0.627	0.602
U-T&T	MiT-B2	0.617	0.590
Ensemble		**0.640**	**0.612**

Table 8. Comparison of individual models and ensemble models on the test set. The ensemble was created from the models with the MiT backbone. Each model was trained with a different seed.

6. Results

In this section, we provide the final setup of the model and the results.

Final setup. We used all the information learned in the previous experiments to train models while using the best setup. All models were trained with the AdamW optimizer, multi-step scheduler with a starting learning rate of 0.0001, batch size equal to 10, focal loss with label smoothing, and utilizing a two-stage training procedure.

Additionally, to maximize the use of data, instead of using 20% data for validation we used only 10% for validation and the rest for training. We trained the final models with three different backbones, where each backbone was trained on three different data folds. The data folds were created by shuffling the sub-regions of data origin and then for each fold 10% different regions were selected for validation and the rest were kept for training. It should be noted that values in Tables 9 and 10 therefore can not be compared to values in other tables and also results between different folds are not comparable.

In Table 9 are the results of U-Net models from the first training stage which were then used to train full models. Results of the full models are in Table 10.

Model	Backbone	Fold	mIoU
U-Net	ResNet-34	0	0.590
U-Net	ResNet-34	1	0.572
U-Net	ResNet-34	2	0.565
U-Net	ResNeXt-50-32x4d	0	0.600
U-Net	ResNeXt-50-32x4d	1	0.583
U-Net	ResNeXt-50-32x4d	2	0.554
U-Net	Mit-B2	0	0.605
U-Net	Mit-B2	1	0.607
U-Net	Mit-B2	2	0.588

Table 9. Results of U-Net models trained on different dataset folds.

Name	Model	Backbone	Fold	mIoU
RsN-0	U-T&T	ResNet-34	0	0.559
RsN-1	U-T&T	ResNet-34	1	0.595
RsN-2	U-T&T	ResNet-34	2	0.601
RNX-0	U-T&T	ResNeXt-50-32x4d	0	0.574
RNX-1	U-T&T	ResNeXt-50-32x4d	1	0.593
RNX-2	U-T&T	ResNeXt-50-32x4d	2	0.597
MiT-0	U-T&T	Mit-B2	0	0.574
MiT-1	U-T&T	Mit-B2	1	0.614
MiT-2	U-T&T	Mit-B2	2	0.612

Table 10. Results of full models trained on different dataset folds.

Model ensemble. To further improve the final results, we tested ensembles composed of the models from Table 10. We abbreviated the names of the models, ResNet-34 is denoted as RsN, ResNeXt-50-32x4d is denoted as RNX, and MiT-B2 as MiT. The number after the hyphen denotes the data fold that was used for the training of the model.

The ensemble output is obtained as the simple average of the individual model outputs. The weighted average did not bring any improvements in our testings. We tried two options for obtaining the final decision: 1. we averaged the logits of individual models and then applied softmax, 2. we averaged the results of the models after applying the softmax function to the output of individual models. Results in Table 12 showed that the first option achieved slightly better results. We believe this is caused by the known phenomenon where the neural networks provide higher values of logits for the inputs for which they have high confidence in their prediction.

We first tested a combination of the same models trained on the different data folds. The best results were achieved by the MiT model. Secondly, we explore possibilities of ensembles created by combination of different models. The best one is reported as Ensemble-03. It is composed of models with three different backbones. It reached comparable results with the best ensemble composed of the same models, nevertheless, its inference was slightly faster. Lastly, we bench-marked ensembles composed of four different models while still holding the constraint of the FLAIR challenge on the maximum inference time. The best results are reported as Ensemble-04 which combined MiT and ResNeXt models. With the Ensemble-04 we reached $mIoU = 0.641$ which is new state-of-the-art and also the best results in the competition leaderboard.

To further analyze the behavior of the final ensemble, we compare test results for individual segmentation classes, see Figure 5. It can be seen that the ensemble improved all class segmentations.

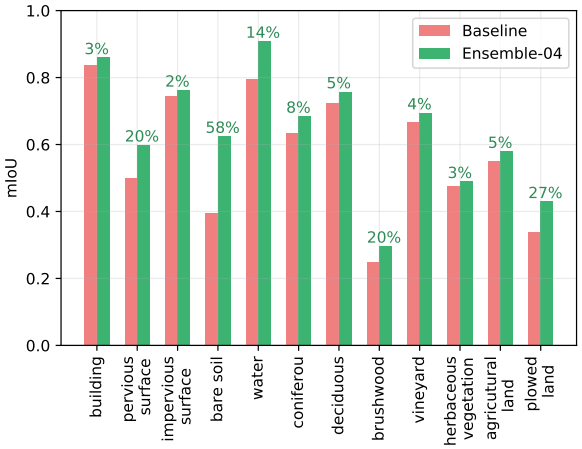

Figure 5. Comparison of test results of the baseline and the final model. The value shown over the bars is the relative improvement.

Inference time. Lastly, we measured the inference time of the two best ensembles. In Table 11 are measurements of the time of baseline and ensembles. Times in each row were measured on the same machine utilizing an NVIDIA Tesla T4 graphic card and 32GB RAM.

Name	Baseline time [s]	Ensemble time [s]	Relative time
Ensemble-03	1267 ±8	2243 ±45	1.776 ±0.035
Ensemble-04	1267 ±8	3024 ±59	2.380 ±0.046

Table 11. Inference times of the final ensemble models.

Summary. In this work, we conducted a series of experiments. Our main observations from the experiments are as follows. The model generally benefited more from backbones with a lower number of parameters. Transformer-based backbone model achieved better performance than

name	Model 1	Model 2	Model 3	Model 4	mIoU logits	mIoU softmax
Ensemble-00	RsN-0	RsN-1	RsN-2	-	0.621	
Ensemble-01	RNX-0	RNX-1	RNX-2	-	0.628	
Ensemble-02	MiT-0	MiT-1	MiT-2	-	0.637	0.636
Ensemble-03	RsN-2	MiT-1	RNX-2	-	0.636	0.635
Ensemble-04	**MiT-0**	**MiT-1**	**RNX-1**	**RNX-2**	**0.641**	**0.641**

Table 12. Results of model ensembles on **test set**.

convolution-based models with a similar number of parameters. Additional more sophisticated augmentations like coarse drop and shift/scale/rotate did not improve performance. Multi-step learning rate scheduler with two 90% learning rate drops and the cosine scheduler without warm-up performed notably better than the constant learning rate. We tested multiple commonly used loss functions for semantic segmentation. The change of loss function from cross-entropy to focal loss with smoothing improved results substantially. We introduced a two-staged training procedure which also improved performance. First was trained U-Net only then the full model was then initialized with the pre-trained U-Net trained further. Lastly, we confirmed that the model ensembles significantly improve results at the cost of computational complexity.

7. Conclusion

In this work we summarized our experiments on the dataset from the FLAIR #2 challenge. The goal of the challenge was to propose a model for semantic segmentation of aerial images with support of other various modalities. We adopted the baseline method proposed by the challenge organizers which is based on U-Net and U-TAE models.

We conducted a series of experiments with a goal not only to improve the final results but also to show the importance of different approach components on the overall performance. In our extensive ablation study, we compare the influence of different model backbones, training schedulers, augmentations, loss functions, and training setups. To reach the best results, we created an ensemble of models trained on different folds of the dataset. We were able to improve the *mIoU* from 0.576 to 0.641, which is a new state-of-the-art result. Moreover, our solution achieved first place in the challenge leaderboard.

In our future research, we would like to explore possibilities of metadata incorporation into the final pipeline and also directly to the training procedure. We believe that this information can help the segmentation models to overcome problems with high inter-class variance by adding the final piece of necessary information to produce a correct decision.

Acknowledgement

Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth, and Sports of the Czech Republic. The work was supported by the University of West Bohemia, project No. SGS-2022-017.

References

[1] Institut national de l'information g´eographique et foresti'ere, https://www.ign.fr. 2

[2] Sentinel hub, https://www.sentinel-hub.com/. 2

[3] Hamed Alemohammad and Kevin Booth. Landcovernet: A global benchmark land cover classification training dataset. *arXiv preprint arXiv:2012.03111*, 2020. 2

[4] Ahmad Alzu'bi and Lujain Alsmadi. Monitoring deforestation in jordan using deep semantic segmentation with satellite imagery. *Ecological Informatics*, 70:101745, 2022. 2

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3

[6] Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Veronica Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36, 2012. 2

[7] Anatol Garioud, Apolline De Wit, Marc Poupée, Marion Valette, Sébastien Giordano, and Boris Wattrelos. Flair #2: textural and temporal information for semantic segmentation from multi-source optical imagery. 2023. 2, 3, 4

[8] Anatol Garioud, Stéphane Peillet, Eva Bookjans, Sébastien Giordano, and Boris Wattrelos. Flair #1: semantic segmentation and domain adaptation dataset. 2022. 2

[9] Vivien Sainte Fare Garnot and Loic Landrieu. Lightweight temporal self-attention for classifying satellite images time series. In *Advanced Analytics and Learning on Temporal Data: 5th ECML PKDD Workshop, AALTD 2020, Ghent, Belgium, September 18, 2020, Revised Selected Papers 6*, pages 171–181. Springer, 2020. 3

[10] Vivien Sainte Fare Garnot and Loic Landrieu. Panoptic segmentation of satellite image time series with convolutional temporal attention networks. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 4872–4881, 2021. 3

[11] 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. 3, 4

[12] Daniel Hernández, José M Cecilia, Juan-Carlos Cano, and Carlos T Calafate. Flood detection using real-time image segmentation from unmanned aerial vehicles on edge-computing platform. *Remote Sensing*, 14(1):223, 2022. 1, 3

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[15] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017. 2

[16] Maitiniyazi Maimaitijiang, Vasit Sagan, Paheding Sidike, Ahmad M Daloye, Hasanjan Erkbol, and Felix B Fritschi. Crop monitoring using satellite/uav data fusion and machine learning. *Remote Sensing*, 12(9):1357, 2020. 1

[17] Hafiz Suliman Munawar, Fahim Ullah, Siddra Qayyum, and Amirhossein Heravi. Application of deep learning on uav-based aerial images for flood detection. *Smart Cities*, 4(3):1220–1242, 2021. 1

[18] Hafiz Suliman Munawar, Fahim Ullah, Siddra Qayyum, Sara Imran Khan, and Mohammad Mojtahedi. Uavs in disaster management: Application of integrated aerial imagery and convolutional neural network for flood detection. *Sustainability*, 13(14):7547, 2021. 1

[19] Krishna Neupane and Fulya Baysal-Gurel. Automatic identification and monitoring of plant diseases using unmanned aerial vehicles: A review. *Remote Sensing*, 13(19):3841, 2021. 1

[20] Huong Thi Thanh Nguyen, Trung Minh Doan, Erkki Tomppo, and Ronald E McRoberts. Land use/land cover mapping using multitemporal sentinel-2 imagery and four classification methods—a case study from dak nong, vietnam. *Remote Sensing*, 12(9):1367, 2020. 1

[21] Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access*, 9:89644–89654, 2021. 3

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 3

[24] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 240–248. Springer, 2017. 5

[25] Xin-Yi Tong, Gui-Song Xia, Qikai Lu, Huanfeng Shen, Shengyang Li, Shucheng You, and Liangpei Zhang. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sensing of Environment*, 237:111322, 2020. 2

[26] Junjue Wang, Zhuo Zheng, Ailong Ma, Xiaoyan Lu, and Yanfei Zhong. Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. *arXiv preprint arXiv:2110.08733*, 2021. 2

[27] Mo Wang, Jing Wang, Yunpeng Cui, Juan Liu, and Li Chen. Agricultural field boundary delineation with satellite image segmentation for high-resolution crop mapping: A case study of rice paddy. *Agronomy*, 12(10), 2022. 2

[28] Ziqi Wang, Tao Peng, and Zhaoyou Lu. Comparative research on forest fire image segmentation algorithms based on fully convolutional neural networks. *Forests*, 13(7):1133, 2022. 2

[29] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 4

[30] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 4

[31] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 3