Landing Zone Specification based on UNet and ResNet Semantic Segmentation and for Autonomous Unmanned Vehicles (UAVs)

Buse Ercik
Universita` degli Studi di Milano-Bicocca
ID: 910371

***Abstract*—The aim of this study is to propose an analysis of landing zone detection approach utilizing UNET Semantic Segmentation model and ResNet with a transfer learning method for Autonomous Unmanned Vehicles. Aerial Semantic Segmentation Drone Dataset involves aerial images with its segmentation masks that represent 24 classes. Aerial Semantic Segmentation Drone Dataset composed of 400 aerial images and 400 masks. In order to conduct a comprehensive process, first of all, dataset was split into train, test, and validation sets. Followingly, data was pre-processed which includes resizing, data augmentation, normalization and tiling. A UNet model is created with a structure of encoder and decoder by using PyTorch. Furthermore, UNet model was trained by utilizing Cross-Enropy Loss, AdamW optimizer and OneCycleR learning rate schedular. Finally, the performance of the model was evaluated by observing training loss, validation loss, training accuracy, validation accuracy and mean Intersection over Union for training and validation sets. Afterwards, for the identification of safe landing areas, a post-processing algorithm is created.**

Unet Link:
https://www.kaggle.com/code/buse98/ercik-buse-final-drone-unet-project

ResNet Link:
https://www.kaggle.com/code/buse98/ercik-buse-resnet-final

1. Introduction

Landing zone specification for Autonomous Unmanned Vehicles is critical due to an efficient and safe deployment in rescue missions and delivery purposes [1]. In order to ensure the safety of AUVs and its surrounding areas, locating accurate and optimal landing zones is crucial. The most significant challenge is precise segmentation of the environment due to diverse environments.

Recent researches have emphasized that deep-learning approaches exhibit significant progress in segmentation of the environment [2]. For instance, encoder-decoder architecture of UNet allows high resolution segmentations. While encoder part of the UNet includes convolutional layer with ReLU activations and max-pooling, decoder part upsamples and concatenates with features of encoder to map segmentation [3].

The objective of this study is to explore the ability of UNet and ResNet structure together with Cross-Enropy Loss, AdamW optimizer and OneCycleR learning rate schedular in the detection of landing zones by conducting pre-processing.

2. Methodology

2.1. Data Pre-processing

Before the implementation of model, data exploration and pre-processing were needed to be performed for having an explicit picture of what the data actually is, and how to best manage it. In this part, preparation and exploration of dataset, resizing, data augmentation, normalization and tiling were performed.

2.1.1. Train-Test-Validation Split

Dataset is composed of 400 aerial images and 400 masks. Masks are annotated to categorize pixels into 24 various classes such as dog, vegetation and other related land types.

In order to achieve an adequate model evaluation, images are stratified into training, validation and test sets. Initially, dataset was split into training and test set with 80:20 split which yields to 320 training images and 80 test images. Hence, test images became the unseen data to test the model. Afterwards, 320 training images was divided into validation and training sets with a split of 75:25. Finally, 240 images for training set, 80 images for validation set, 80 images for test set were obtained.

2.1.2. Resizing, Data Augmentation, Normalization and Tiling

In order to handle loading and pre-processing data, ‘DroneDataset’ class created. ‘DroneDataset’ class consists of resizing, data augmentation, normalization and tiling.

Resizing, data augmentation, and normalization play an essential role to train robust and trustworthy machine learning models. For the preparation of the dataset before UNet training, resizing, data augmentation and normalization were applied.

Training and validation images were resized into 704 x 1056 pixels. The reason behind the high pixel resizing was to ensure the details of the original images are maintained. This high pixel resizing was important for accurate identification of small features. Also, in order to preserve label integrity, nearest-neighbor interpolation was applied. This strategy prevented introduction of new pixel values which could have given rise to alter class labels.

Mimicking of real-world situations such as lighting variations, different positions of AUVs, various noise levels, different spatial diversity are important for model generalization on unseen data. In order to perform data augmentation, vertical and horizontal flip, grid distortion, brightness and contrast modifications, and gaussian noise were implemented for the enhancement of the robustness of model. Employment of those augmentations avoids overfitting and creates effective performance under different environmental conditions.

Another important pre-processing step is normalization. Pixel values were scaled by the employment of the subtraction the mean and then division the standard deviation for each pixel. Hence, each pixel values have a mean of zero and a standard deviation of one. This process, enables consistency which results fast convergence and the stabilization of learning process.

Additionally, a tiling function was benefited for the division of images into small patches like 512x768 pixels. Due to the fact that the existed memory has limits and also dataset consists of detailed images with high resolutions, tiling is an enlightening way to manage memory in an effective way. With the help of this method, model focuses on details.

For the validation set, only resizing, horizontal flipping, and grid distortion was applied. More straightforward method was implemented to the validation set without introducing extreme variability to make validation-set more realistic evaluation.

In the 'DroneDataset' class, images and its corresponding masks are retrieved depending on their names of files created by appending '.jpg' or '.png' to the base file name. Afterwards, pre-processing was applied. Then, image and mask pairs were become PyTorch tensors for model training.

2.2. U-Net Structure

U-Net was implemented to obtain pixel-level classification task. Model has an encoder and a decoder part. Initialization of the model was performed with 3 input channels which are RGB and output classes which are 24. Encoder part of the model composed of downsampling blocks that increases feature depth from 64 to 1024 and halves spatial dimensions. Blocks involve two convolutional layers with batch normalization and ReLU activation, max pooling, and dropout. Max pooling provides the reduction of spatial dimension while dropout prevent overfitting with regularization. Moreover, in the middle of downsamplin and upsampling, there is a bottleneck which captures the most abstract features and it was composed of two convolutional blocks. In the upsampling (decoder) part, blocks composed of transpose convolution, double convolutional block and dropout which decreases feature depth from 1024 to 64 and doubles spatial dimensions. Upsampling blocks use features to turn back to the original resolution. The last layer is a convolution which maps 64 features to the 24 target classes. The final layer does not include an activation function due to the fact that CrossEntropyLoss includes softmax activation function and softmax was used during training.

2.3. ResNet Transfer Learning

ResNet-50 was utilized to extract features from aerial images. Pretrained weight of ResNet-50 on the ImageNet dataset was used. ImageNet dataset includes 1.2 million images and 1000 classes. Model composed of 50 layers with convolutional layers, batch normalization layers, and residual blocks. Inital layer was designed with a convolutional layer which includes 7x7 kernel size, 64 filters and max-pooling layers. Followingly, 4 core blocks which are residual blocks was implemented. Each block composed of several convolutional layers which also includes shortcut connections in order to mitigate vanishing gradient. Each residual block halves spatial dimensions and doubles the number of filters from 64 to 512. As a final layer, average pooling layer was implemented which decrease feature map to a single value. Final fully connected layers were removed in order to reduce the depth from 2048 to 1024 and followingly second convolutional layer was added to map features to 24 classes. Moreover, an upsampling layer was implemented to increase spatial resolution to match input size.

2.4. Model Training and Evaluation Metrics

Model training consists of numerous critical steps to provide an optimal performance and learning. Forward propagation was the initial point to start. Model analyzed input images and generate predictions. Followingly, predicted masks and true masks were compared between each other by utilizing Cross-Entropy Loss. Cross-Entropy Loss employs a softmax layer in order to transform output scores of the model into a probability distribution across classes. After, the predicted probability and actual class labels differences were computed. This computation denoted how well the match between true and predicted labels.

Followingly, in order to calculate the gradients of the loss regarding parameters of the model, backpropagation ws performed. Calculated gradients indicated how parameters of the model ought to be changed for the minimization of loss. AdamW was utilized for the update of parameters with the help of gradients. AdamW parameters which are learning rate and weight decay were adjusted for the adjustment of step size for the model's parameters depending on gradient. Weight decay was applied because it inhibited the excessive weight values which enabled to prevent overfitting. This mechanism allows larger adjustments for sparse gradients and lower adjustments for larger gradients which provides an effective convergence.

For the enhancement of training, OneCycleLR schedular was utilized for a dynamic modification of learning rate in a cyclical way across epochs. This adjustment prevents potential stuck in local minima by the initialization of learning rate at low level, after an increase to peak and a decrease again to a low level. This technique encouraged more exploration.15 epoch training was performed with mini-batch size of 3. This arrangement made computationally efficient training possible.

Two primary evaluation metrics were performed which are pixel accuracy and mean Intersection over Union (mIoU). Pixel accuracy calculates the division of the number of correctly predicted pixels to the total number of pixels.

$$\text{Pixel Accuracy} = \frac{\text{Number of Correctly Predicted Pixels}}{\text{Total Number of Pixels}}$$

Formula 1. Pixel Accuracy Calculation

Moreover, mean Intersection over Union was calculated which is widely used in segmentation models. It is a measurement technique that calculates the overlap between predicted mask and ground truth mask. Afterwards, mean of all classes IoU computed.

$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}} = \frac{|\text{Predicted} \cap \text{Ground Truth}|}{|\text{Predicted} \cup \text{Ground Truth}|}$$

Formula 2. Intersection over Union Calculation

At the end, the model output was utilized for the identification of a safe landing zone. In the designed methodology, first of all, the trained models were run to a set of test images. Afterwards, predictions were moved to CPU for further examinations. Then, a dictionary was created to link the class numbers to the descriptive names of classes.

For the identification of safe zones, some classes mapped as safe and some classes mapped as unsafe. Followingly, binary masks were produced and safe areas are detected by determining whether a pixel value match safe class or not. Then, binary mask was used to figure out the safe landing areas contours. 500 pixels were predefined as a threshold to land the AUVs in a safe place. Optimal landing location was detected by the calculation of the greatest safe area and the landing coordinate was in the middle of that bounding box.

3. Results and Discussion

At the initial stage of the study, pre-processing part was only included 128x128 resizing, horizontal flipping and normalization. After the training 3 epochs, it was clearly seen that this methodology was too inadequate and simplistic for the the training process. Model showcased validation accuracy stuck at 0% meaning the inability of generalization whereas accuracy score of training set began at 90% and reached to 100% in 3 epochs. While those results revealed severe overfitting, existing GPU resources ran out. In order to overcome those circumstances, several experimentations including training with various learning rates and batch sizes, different dropout percentages and plenty of modifications to the U-Net structure including decreasing and increasing layers were processed. Any satisfactory result was obtained. Consequently, tiling technique was applied and enhancement of data augmentation was processed including resizing images to 704x1056 pixels, application various flips, brightness and contrast adjustments, adding Gaussian Noise and grid distortion. As result of these adjustments, more stable training outcomes were obtained. Each epoch approximately took 4minutes and due to the GPU limitations only 15 epochs were trained. Likewise, on account of the GPU limits only two models' comparison was performed which are U-Net and ResNet

Figure 1. and Figure 2. exhibits the illustration of Intersection over Union results both for validation and training sets. It is seen that training shows an upward trend which denotes that the trained model was learning how to segment images in a more accurate way after each epoch. Even though the validation IoU shows a fluctuation, it exhibits an overall improvement. That fluctuation may be caused by the relatively a smaller number of images in validation set and the variability in the validation data. Furthermore, early strong performance of the validation set can be reasoned by the first model fit better to validation data because of the randomness.

Figure 1. mean Intersection over Union results of U-NET

Figure 2. mean Intersection over Union results of Res-Net

It was clearly seen that pre-trained Res-Net model has higher mIoU(0.14) than U-Net mIoU (0.35).

Figure 3. and Figure 4. which present training accuracy and validation accuracy scores in 15 epoch reveals that higher than 50% accuracy score in first few epochs of U-Net and Res-Net was quickly reached for training and validation sets. This means that in the first few epochs loss was minimized and weights were adjusted by model in a quick way which gave rise to a gain in accuracy. Moreover, fluctuations were observed in validation set like it was seen at mIoU for both Res-Net and U-Net model, this can be again caused by data variability and learning rate adjustments which was performed by OneCycleR scheedular.

It can be observed from the accuracy results that the pre-processing modifications helped the model to generalize better and prevent overfitting for both models.While training and validation accuracy results increased to around 60% for U-Net, ResNet striked 80% accuracy. Since ResNet used as a Transfer Learning method and its pre-trained weights used, ResNet has the highest accuracy compared to U-Net.

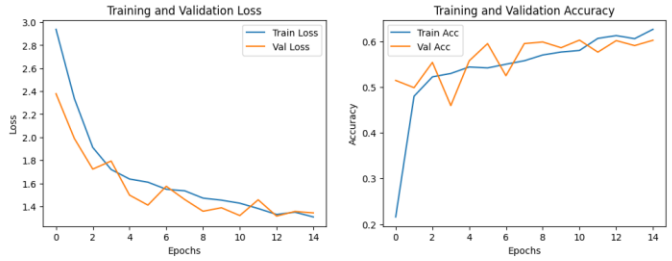

Figure 3. Loss and Accuracy Graphs of U-Net Model

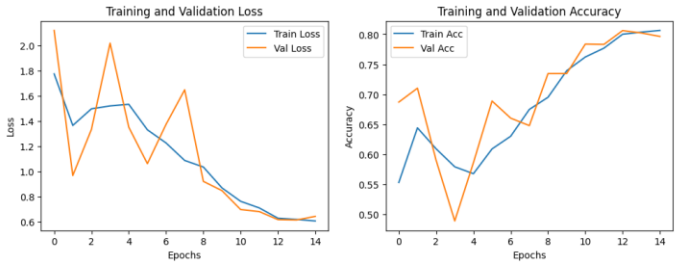

Figure 4. Loss and Accuracy Graphs of ResNet Model

Furthermore, loss results of training and validation sets for both ResNet and U-Net showcased a downward trend over epochs which demonstrates that predictions of the model less erroneous across time. At the end of the training while ResNet loss decreased to 0.6, U-Net loss diminished to 1.4.

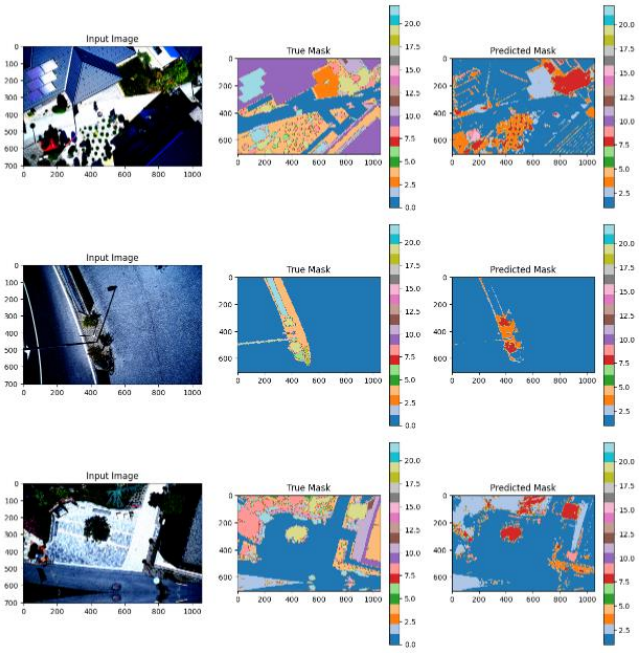

Figure 5. Input images, True masks and Predicted masks visualizations of U-Net

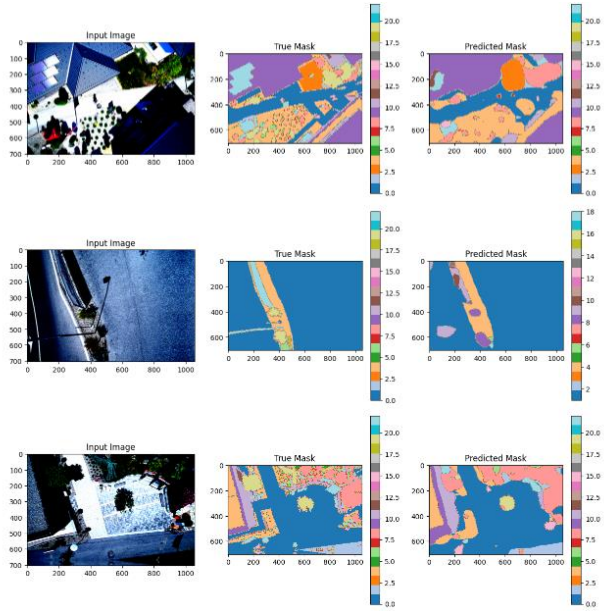

Figure 6. Input images, True masks and Predicted masks visualizations of ResNet

Figure 5. and Figure 6. illustrates the qualitative results which represents inputs, ground truth and predictions. Predictions of the model match with ground truth masks in a good way. Also, both models can correctly detect classifications which is critical for the identification of safe zones for landing. More detailed segmentation was performed by U-Net model while Res-Net predictions were not detailed as U-Net outputs. This is caused by couple of reasons that U-Net preserves spatial information because of its structure includes skip connection which combine features from encoder with the features from decoder while ResNet uses much more straightforward decoder

Figure 7. and Figure 8 exhibits safe areas for landing as a red dot mark. Safe and risky zones were correctly classified by categorizing regions.

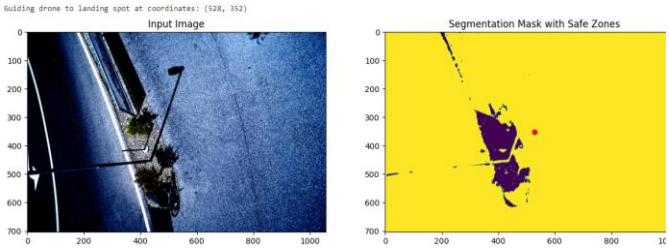

Figure 7. Landing zone marked as dot mark by U-Net

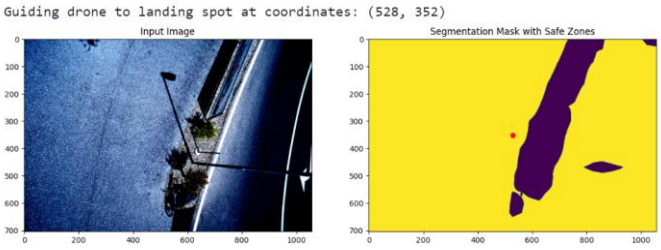

Figure 8. Landing zone marked as dot mark by U-Net

In summary, ResNet outperformed U-Net with its superior performance in segmentation. It highlighted higher accuracy and mean Intersection over Union. This made ResNet preferred model for a 15-epoch training. It was obvious that its robust performance was caused by its pre-trained nature which made generalization well with limited epoch possible. On the other hand, U-Net performed a much more detailed semantic segmentation. It has a sensitivity to small variations which sometimes causes misclassifications. Due to the current limitations of GPU, the exploration of other models and much more extensive training time were restricted.

4. Conclusion

In this study, an analysis of the landing zone detection for Autonomous Unmanned Vehicles by utilizing Semantic Segmantation masks of U-Net and ResNet with transfer learning was performed. Pre-processing of the Aerial Semantic Segmentation Drone Dataset was performed. Followingly, UNet and ResNet with transfer learning was implemented by using Pytorch. Models were trained with Cross-Entropy Loss, AdamW optimizer and OneCycleLR learning rate schedular. Performances of the models were evaluated by training and validation loss, training and validation accuracy and mean Intersection over Union calculation. Subsequently, in order to detect safe landing areas, a post-processing algorithm was implemented. Despite the detailed outputs of U-Net, ResNet performed superior to U-Net with respect to higher accuracy and mIoU. On account of the pre-trained weights were utilized for ResNet, the model became more effective one. Additional data and training may improve both models. Also, variety of deep-learning models can be trained.

References

[1] M. W. Khan, S. H. H. Shah, and M. Sajjad, "A Comprehensive Survey of Unmanned Aerial Vehicle Uses and Applications," Journal of Mechanics of Continua and Mathematical Sciences, vol. 15, no. 4, pp. 136-145, 2020.

[2] X. Li and Y. Liu, "Deep Learning-Based Maritime Environment Segmentation for Unmanned Surface Vehicles Using Superpixel Algorithms," Journal of Marine Science and Engineering, vol. 9, no. 12, pp. 1329, 2021.

[3] F. Ekinci, T. Asuroglu, and K. Acici, "Deep-Learning-Based Approaches for Semantic Segmentation of Natural Scene Images: A Review," Electronics, vol. 12, no. 12, pp. 2730, 2023.