U-net-based Early Wildfire Detection with Fastai

Linhan Qiao[1], Youmin Zhang[2*], Yaohong Qu[3]
[1]*Department of Mechanical, Industrial and Aerospace Engineering, Concordia University*, Montreal, Canada
742954173@qq.com
[2]*Department of Mechanical, Industrial and Aerospace Engineering, Concordia University*, Montreal, Canada
ymzhang@encs.concordia.ca*
[3]*School of Automation, Northwestern Polytechnical University*, Xi'an, China
qyh0809@126.com

***Abstract*—This paper purposed a scheme of building U-net with Fastai framework for early wildfire detection, which could be deployed on ground work station. The designed model in this paper works especially for smoke detection. This U-net model is based on Resnet34, and attention gates are applied on skip connections through Fastai. Different from Tensorflow or Pytorch, Fastai is a higher level API, by which the transfer learning could be easier and faster deployed for network tuning. The data used in this paper is mixed by data-sets from Kaggle and experiment captured images. The experiment segmentation results show that, after unfreeze the entire encoder-decoder structure, the accuracy is acceptable 94.9%, the attention gates helped reducing the impacts of noisy data and improved the generalization ability of this model.**

***Index Terms*—wildfire detection, U-net, Fastai, Attention gate**

I. Introduction

Wildfire is threatening forest resources a lot and it is also considered as a disaster to the life safety of animals and humans who live in or nearby forests. Commonly, the wildfire could be considered as an abnormal state of forest or grass environment system. Because of the complicity of decoupling the flame action and environmental varieties, such as vegetation, wind action, topography during the process of wildfire spreading, it is necessary to detect and manage the wildfire at its early period. Discovering and detect wildfire early is demonstrated as one of the most efficient and demanded schemes for preventing or managing wildfire to decrease losses. Comparing with filtering schemes or other kinds of sensors, the RGB cameras cooperate with infrared camera could capture the image or heating information in forest more timely for recognising wildfire. Encouraged by the development of the study of neural network models and computer vision (CV) technology, computational intelligent image-based or video-based wildfire detection is now being widely accepted and applied in forest surveillance and forestry study area. [1].

U-net, one of these typical NN-based segmentation models, is an encoder-decoder structured model for image segmentation or object detection [2]. Especially, for some medical application, U-net is considered as one of the most efficient model to use the Graphics processing units (GPUs) memory. Also, different level features are extracted from multi-scale, which means that smaller dataset could be used to train an acceptable segmentation model. Commonly, in wildfire detection, the data of forest environment is not easy to acquire, and the available data are less. Therefore, U-net is recommended to detect wildfire in early period. To be more specific, U-net is appropriate to detect or segment smoke during early wildfire as the non-regular shape smoke raise before flames are visible [3].

Based on these analysis above, this paper proposed a U-net based smoke detection scheme using Fastai framework. The rest part of this paper is arranged as follow: In Section 2, related works of U-net and smoke segmentation are briefly reviewed. The methodology of U-net based wildfire detection is stated in Section 3. Details of the design of experiments and tests are arranged in Section 4. Finally, conclusions and potential future works are stated in Section 5.

II. Related Works

With the great development of the study of neural network (NN) or even deep neural network (DNN) models, there are many studies consider NN for smoke segmentation because of their faster convergence and deployment through the supports of acceleration of matrix computation (development of GPU). Commonly, it is illustrated that three typical models could perform well for segmentation tasks: Non-end-to-end model, end-to-end model and sequence-to-sequence model.

Non-End-to-End model: Generally, region-based convolutional neural networks (R-CNNs) are considered as Non-End-to-End model for detection tasks. At first, this kind of models extract region proposal areas where the targets are suspected located before input. Then, these proposal areas are loaded into CNNs to detect whether the targets are contained to finish the detection missions. Faster R-CNN is one typical and popular Non-End-to-End model for wildfire detection [4]. The Advantages of such schemes could be illustrated that the parameters could be shared, and the detection accuracy could be increased a lot. Based on the concept of region proposal, a bounding boxed of the target area will be output as the proposal region area are loaded into the model [5]. You-look-only-once model (YOLO) achieved faster and lighter-weight for simple detection tasks [6]. However, the performance of YOLO for multiple no-rigid body, such as multiple smoke area over flames in forest scene, is still facing some challenges.

End-to-End model: Compared with the region proposal scheme of Non-End-to-End models, End-to-End models could

be considered as a process in which the neural network model learns to extract feature itself. Because the feature extraction or feature learning all happen inside the model after masks or labels are supplied, one of the salient advantage of End-to-End scheme is less emphasis on the feature itself, which decreased the faults of manually feature extraction. Fully connected networks (FCNs) is consider for wildfire detection early [7]. Based on these models, some schemes are developed by focusing on increasing the accuracy or training speed [8]. And in recent years, different structured models are designed for better wildfire detection performance. A 3D-fully connected pyramid classification is designed in [9] to deal with the false-positive problem when detecting wildfire smoke. Encoder-Decoder structured models, like U-net could also be considered as a kind of FCN for wildfire detection. U-net is able to be developed from medical applications into the wildfire detection or forestry applications. In [10], attention gate (AG) enhanced U-net and squeezeNet modules [3] are applied for fire or flame detection. However, that model only classifies flame, does not consider the smoke of early wildfire period, and the detection accuracy is around 81%, which still need to be improved

Sequence-to-Sequence Model: Some of the recurrent neural networks (RNNs), such as long-short-term memory (LSTM) model [11], could be defined as Sequence-to-Sequence model and have proved their performance in wildfire detection. Because of the re-hot of attention mechanism, there are also many schemes prefer to apply attention layers with their model for potential future works [12]. However, performance of models with pure attention mechanism and some attention enhanced network models such as vision transformer [13] still need more time and demonstration.

III. Methodologies

In this section, the methodology of building a U-net with Fastai for wildfire detection is stated. The Fastai framework will be introduced first. Then the design of U-net detection could be separated into Data source, U-net modeling, training, and testing. Then it is made as production.

Fastai is a Pytorch-based API which provides high-level components that can quickly and easily provide state-of-the-art results in standard deep learning domains, and provides low-level components that can be mixed and matched to build new approaches [14]. Therefore, some convenient methods are supported in fastai to help build neural networks for specific missions.

It is illustrated in Fig.1 that Fastai consists different level of APIs. Low level APIs and high level applications are all contained in Fastai to help meeting design goals. For the objective of this paper, it is possible to create a state-of-art vision model using transfer learning to segment smoke in forest scene so that the wildfire could be detected early through forestry monitoring systems, such as watch towers and especially the unmanned aerial vehicles (UAVs).

Because of the advantage of U-net that it does not require large data set. And the short skip-connections of U-net insured to capture enough original features with limited numerous of data. U-net is appropriate to detect wildfire smoke and flame. There are also some optimization for U-net to improve its performance. In recent research, attention mechanism is getting hot to optimize or develop the CNN models [10], attention gate could be designed on skip-connections to deal with the false-positive detection problem, and the model is also possible to be squeezed as a smaller model to work on-board.

As analyzed above, smoke image segmentation is considered as an efficient scheme for early wildfire detection, because the smoke could be much earlier viewed than flame. By analyzing these evidences, some characters of U-net make it more suitable for early wildfire detection or wildfire smoke segmentation task. They are stated as follow:

Encoder-Decoder structure: The U-net is typical encoder-decoder structure network which connects sub-sampling processes to up-sampling processes through skip connections. The skip-connection of U-net helps to avoid the size of image shrink too much. It could also be thought as a process of multi scaled feature mixture, where the features are added pixel by pixel, and the concatenation of feature maps [15].

Multi-scaled feature extraction: The real wildfire smoke are often not that easy to get, the size of data set for training is often limited. This reason caused the low level features of original images very important to train the network. Multiple layers of down-sampling of U-net could ensure to capture most low-level features. On the other hand, the flame and smoke are non-rigid body, they do not have stable and clear edges. The low resolution and high resolution process of U-net can help to understand the edge of smoke to segment them earlier.

Therefore, U-net is one of the most appropriate NN-based schemes for wildfire detection. With the help of Fastai, it is more simple to build up a U-net and optimize it with attention gate to finish the smoke detection task.

The U-net model in this paper is based on Resnet34 , it could be illustrated in Fig. 2. The proposed structure in this paper is based on the first stream FCN of 'two-stream' model which is stated in [8]. It ensured the model extract enough features from inputs. Between the Convolution and ReLU activation, there is a Batch-Norm to averaging the batch height and weight by channel [16].

Because of the challenge of smoke detection that small objects with large shape variability leads to higher false-positive predictions of. Attention gates is applied at skip-connections of U-net model. The skip connection in U-net is similar as the residual connections in ResNet, where they supply residuals between local layers and the connected layers. However, in U-net, the skip connections are adding the cropping results of the sub-sampling part layers to up sampling part layers, where crop functions cropped the encoder part feature maps into the same size matched the decoder part ones. The application of attention gates at skip connections could be illustrated as Fig. 3. As shown in Fig. 3, gating vector $g_i \in \mathbf{R}^{F_g}$ for each pixel i contains contextual information to prune lower-level feature responses, and addictive attention could be applied to get the gating coefficient. Balanced the squeeze module

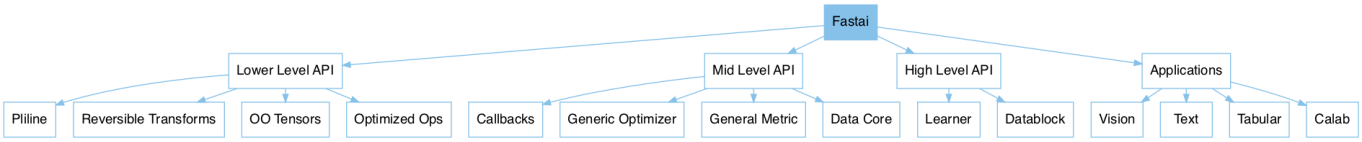

Fig. 1: The layered API of Fastai [14]

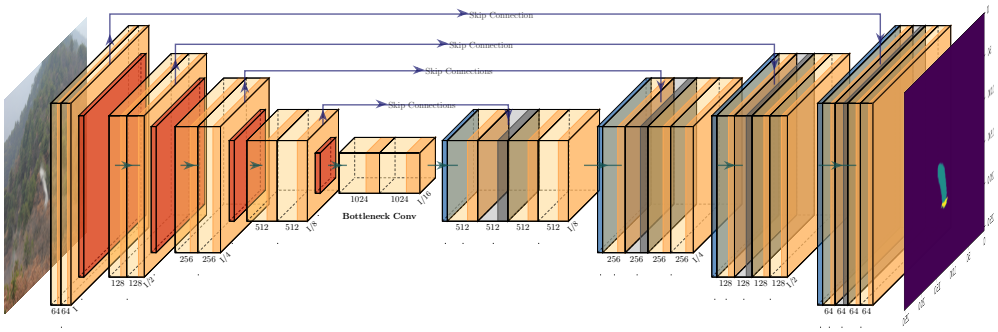

Fig. 2: Schematic diagram of the structure of U-net for wildfire smoke detection

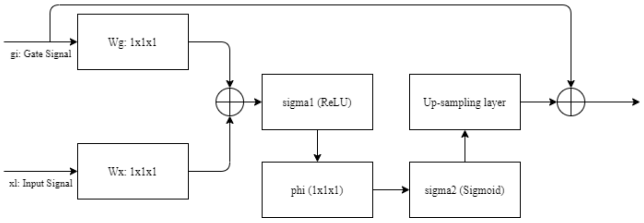

Fig. 3: Brief process of attention gate

significantly decreased the computing complexity, addictive attention sacrificed a small computing resource to achieve higher accuracy. The addictive attention could be formulated as Eq. 2

$$q^l_{attention} = \phi^T(\sigma_1(W_x^T x_i^l + W_g^T g_i + b_g)) + b_\phi \quad (1)$$

$$\alpha_i^l = \sigma_2(q^l_{attention}(x_i^l, g_i; \bigtriangledown)) \quad (2)$$

where σ_1 and σ_2 separately remain the activation of ReLU and sigmoid function, $\bigtriangledown$ consists of a set of AG parameters: linear transformations $W_x \in \mathbf{R}^{F_l \times F_{int}}$, $W_g \in \mathbf{R}^{F_g \times f_{int}}$ and $\phi \in \mathbf{R}^{F_{int} \times 1}$; bias terms $b_g \in \mathbf{R}$ and $b_\phi \in \mathbf{R}$. α_i^l is acquired by linearly mapping concatenated features x and g to a $\mathbf{R}^{F_{int}}$ dimensional space.

Information extracted from coarse scale is used in gating to disambiguate irrelevant and noisy responses in skip connections. And AG filter the activation of neurons during forward pass and backward pass. It enhanced the parameters of shallow layers to update based on special regions. It could also be said that only crucial features are paid attention (Why it is called attention gate). The update rule in layer $l-1$ could be formulated as Eq. 3

$$\begin{aligned}\frac{\partial(\hat{x}_i^l)}{\partial(\Phi^{l-1})} &= \frac{\partial(a_i^l f(x_i^{l-1}; \Phi^{l-1}))}{\partial(\Phi^{l-1})} \\ &= a_i^l \frac{\partial(f(x_i^{l-1}; \Phi^{l-1}))}{\partial(\Phi^{l-1})} + \frac{\partial(a_i^l)}{\partial(\Phi^{l-1})} x_i^l\end{aligned} \quad (3)$$

where x^{l-1} remains the feature map in layer $l-1$, i is the spatial subscript, the function $f(x^l - 1; \Phi^l - 1) = x^l$ applied in convolution layer l is characterised by train-able kernel parameter $\Phi^l - 1$, a_i remains the attention coefficients which identity salient image regions and prune feature responses to preserve only the activation (ReLU, Sigmoid) relevant to the specific task.

The negative cross entropy is chosen as the loss function of U-net,

$$E = -\sum_{x \in \Omega} w(x) \log(p_{l(x)}(x)) \quad (4)$$

where x is the positions of pixels in a whole image; $p_{l(x)}(x)$ represent the possibility of real mask label output on feature channels (final output 3); $w(x)$ is the weight to emphasis more on the boundaries between objects to detect. Therefore, the output pixel-wise softmax is

$$p_k(x) = \exp(a_k(x))/(\sum_{k'=1}^{K} \exp(a_{k'}(x))) \quad (5)$$

where $a_k(x)$ is the value of pixel x on channel k of output layer; K is the number of classes. Then, the output $p_k(x)$ of Eq. 5 is the possibility of x belonging to class k.

IV. Data-set and Experiments

The data used in this paper and the process of training and fine-tuning the U-net is stated in this section. Then, there is

the analysis of the training and testing outputs and prediction results.

A. Data

The data used in this paper could be separate into two parts. The first part is an open data source from 'Kaggle'. These smoking pictures are captured by watchtower or UAVs. The second part of data comes from an experiment designed by team of prof.Qu in October 2020, In this experiment, the smoking images are captured by DJI Phantom 4. These two parts data are shuffled and labelled through pixel region of interest (RoI) labels by Image-Labeller App of MATLAB. By which, mask labels are handle-made.

It is convenient for Fastai to accomplish data augmentation by `tfm`, which could transform the data into different sizes, angels, colors, or even change the brightness, blur them. The augmentation example is shown below.

```
from fastai.vision.all import *
data = (src.transform(get_transforms(),
        size=size,
        tfm_y=True).databunch(bs=bs)
        .normalize(imagenet_stats))
'''
In data bunch, 'bs' is the batch size of
input data.
'''
```

The input images used in this paper are resized into (255, 255), and normalized based on the standards of Imagenet.

B. U-net Deployment

The U-net could be deployed through the `unet_learner` of `learner` of Fastai. The input to the `unet_learner` includes data, built U-net model, and evaluation ratios. After training, the model could be saved through `learner.save` as a product.

C. Training

Fastai is helpful to faster fine-tuning the DNN models such as U-net. In this paper, simulated fire annealing is applied for finding the learning rate. With the support of CUDA, half-float precision (FP16) is applied for saving computational resource or Ram [17]. With the help of Fastai, the learning rate could be find through Simulated annealing algorithm (SAA) [14]. As SAA is applied, the learning rate will increase in a slope of 1, and then getting down slowly. This trick helps to find an appropriate learning rate before the formal training process. As shown in Fig. 4, it find the appropriate learning rate where the convergence happens most quickly. Based on experiences, the learning rates could often be chosen from $1e^{-6}$ to $1e^{-4}$. It is shown in Fig. 4 that the appropriate learning rate of the U-net for processing these data is $1.58e^{-4}$.

There are 415 images used as data set for the training, and the model is trained for no more than 20 epochs for avoiding over-fitting. The accuracy of each epoch of the frozen training process is shown in Table. I. It is shown that the losses are decreasing as the accuracy is increasing. Fig. 5 shows the changing of the training loss and validation loss. It could be seen that the losses are both decreasing and the slope of validation loss is finally near zero, which means the model is trained enough, it is time to stop the iteration, or there would be over-fitting.

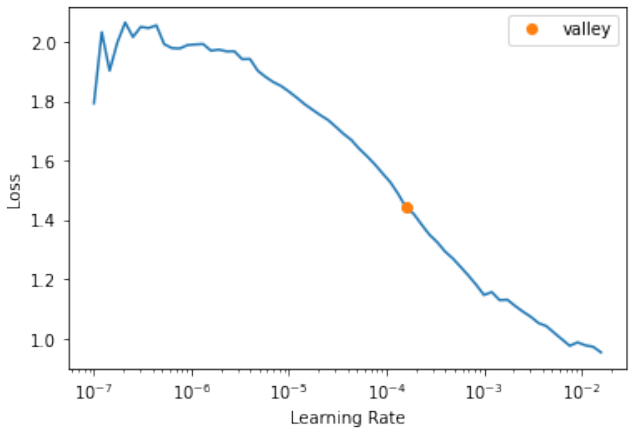

Fig. 4: Finding learning rate through the convergence of loss using Fastai.

TABLE I: U-net performance in (frozen) training process

Epochs	Training Loss	Validation Loss	Accuracy
0	0.273335	0.259231	0.900746
1	0.194220	0.249152	0.905565
2	0.168197	0.157966	0.937570
3	0.150668	0.149131	0.936609
4	0.132787	0.162676	0.939816
5	0.124933	0.145369	0.940570
6	0.110619	0.131302	0.945522
7	0.087399	0.139759	0.945217
8	0.080993	0.142181	0.945206
9	0.079191	0.129560	0.946445
10	0.068988	0.130467	0.947458
11	0.070173	0.131648	0.947665
12	0.062087	0.131910	0.948286

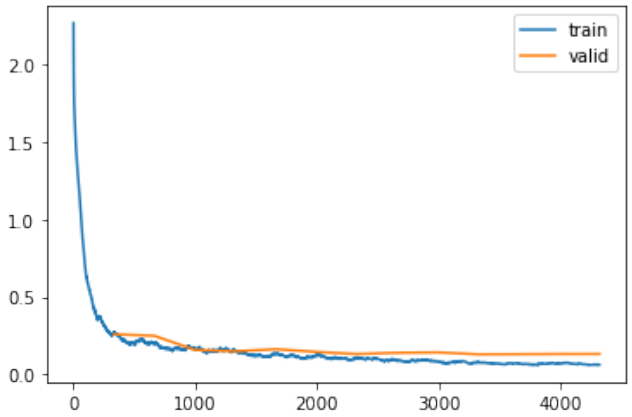

Fig. 5: The trend of the training loss and validation loss in training (frozen) of the U-net model.

Unfreezing the model, there are more parameters of up-sampling part to be know through training. Therefore, training could be carefully continued for some more steps to increase the performance. For the entire encoder-decoder structure, a trick of half-float computing [18] could be applied, which is helpful in saving the computation resource. Finally, the fine-tuned U-net parameters could be saved for testing and product it.

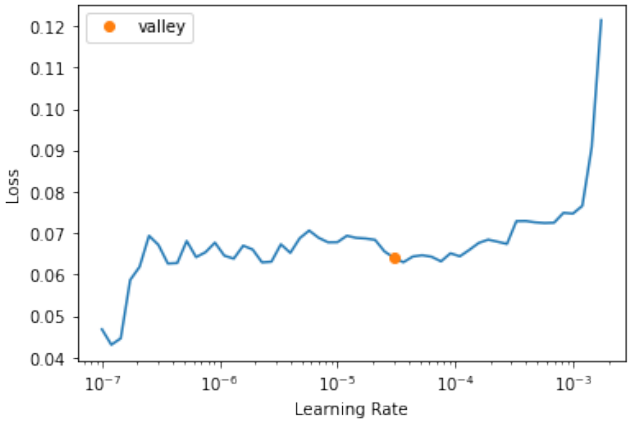

Fig. 6: Finding learning rate through the convergence of loss as decoder is opened.

Fig. 6 is the SAA suggested learning rate at the valley of $3.02e^{-5}$. It is hard for human to find the slope where the loss decreases fastest. The training continued for 4 more epoch and it is get the result of Table. II and Fig. 7. And in Fig. 7, it could be seen the validation loss is starting to have the trend of increasing. The reason might be that the data set is not that big, and all features are nearly extracted or learned by the U-net. With the help of the Graphics processing unit (GPU)

TABLE II: U-net performance in (unfreeze) training process

Epochs	Training Loss	Validation Loss	Accuracy
0	0.063009	0.134996	0.946347
1	0.076282	0.132929	0.946923
2	0.066604	0.127009	0.948871
3	0.062724	0.129913	0.949158

of GTX1660s (6GB), it takes about 35 seconds for every epoch computation in frozen training part and 41 seconds for every epoch computation in unfreeze training part. The final accuracy is 94.916%, which is acceptable for wildfire smoke and flame detection. So, such trained model is saved for testing.

D. Testing and Results

The prediction segmentation results of our model is illustrated in Fig. 8. The figures of first row are the target masked images, and the figures in bottom row are the prediction results. Specially, in the first column, it could be seen that the detected flame is even more specific than the training mask.

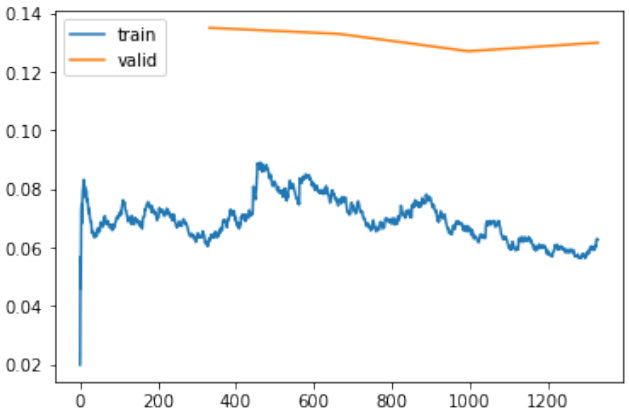

Fig. 7: The segmentation result of front part U-net (unfreeze).

Fig. 8: Prediction results of the trained U-net model. The figures in top row are the segmentation targets; The figures in bottom row are the prediction results.

Roughly area of smoke is located and segmented, it is shown in the last column result.

The prediction performance comparison between our model and pure U-net is shown in Fig. 9. The figures in top row are the original images, the ones in mid row are the predictions of pure U-net fine-tuned through Fastai. The figures of bottom row are the predictions of the model of ours. In Fig. 9, the last two columns are real experiment images and prediction results, where there are only smoke cakes ignition but not real vegetation burning. However, in the third column, it shows that both our model and pure U-net consider the bottom part of the segmented smoke is suspected as flame, which means the feature learning of the model is not only limited in learning the color feature. In the last column result, it is found that the pure U-net prefer to advocate the yellow color leaves under daylight to be flame, which is false-positive prediction. The prediction of our model shows the correct result, it did not predict the leaves to be flames, which is a testimony of the statement that the self-attention mechanism helps to reduce the false-positive predictions for smoke and flame.

On the other hand, it means our model is performing better on the aspect of generalization and robustness. The prediction result is not impacted by the 'changing' of application environment, because the model is mainly trained by the 'Kaggle' data set and few experimental data, but tested on most experimental data and few 'Kaggle' data. And the result is acceptable even there are some noisy images where might be the flame like

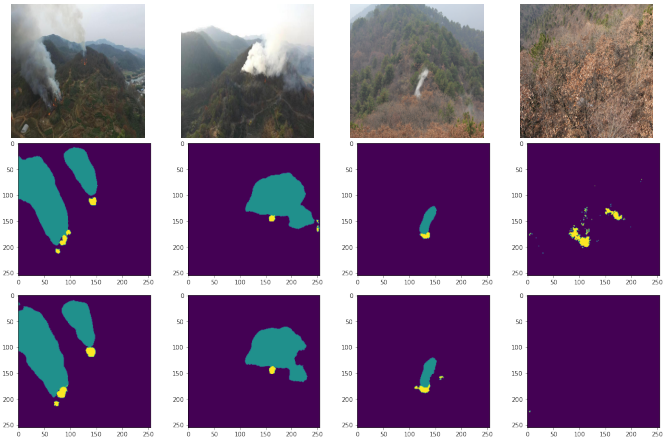

Fig. 9: Comparing prediction results. The figures in top row are the original input images; The mid row are the predictions of pure U-net; The bottom row are the predictions of our model.

colors or shapes, such as the yellow color leaves under daylight in the last column.

V. Conclusions and Potential Future Works

In this paper, a U-net enhanced with self-attention layer using Fastai is proposed for early wildfire smoke and suspected flame area segmentation. Simulated annealing algorithm and half float computation is applied as tricks in the fine-tuning of this model. Through single GTX 1660 GPU (6GB), training time cost is acceptable with 415 training images. The segmentation result of this model demonstrated its robustness generalization ability. The prediction performs better than pure U-net and it learned the feature of suspected flame area. The segmentation accuracy is around 94.9%.

Because of the lack of real wildfire smoke image data, it is more important to training the segmentation model with limited data set. Such U-net-based schemes are considered more appropriate for wildfire detection. And the attention mechanism helped to reduce the false-positive predictions for smoke detection.

Therefore, it could be concluded that the main contribution of this paper is the decreasing of false positive prediction through attention mechanism, fine-tuning time saving through SAA of Fastai and computing acceleration through GPU half float computing.

However, in this paper, the video-based data is theoretically working with this model but not tested. Some of characteristics of crop function are not discussed. Therefore, the wildfire location online need more discussion. That might be the most interested part in potential works.

VI. Acknowledgement

The work of this paper is partially supported by the Natural Sciences and Engineering Research Council of Canada, National Natural Science Foundation of China (No. 61833013), and the Key Research and Development Program of Shaanxi Province of China (No. 2020SF-376).

Thanks to the selfless contributions of Pytorch, Fastai, and Kaggle teams. And specially, thanks to the hard work of the team of Prof. Yaohong Qu of Northwestern Polytechnical University, Xi'an for the data and experience support.

The original data set from Kaggle is available at the link: https://www.kaggle.com/kutaykutlu/forest-fire.

The processed data set, mask label and coding of this paper are uploaded at Github link: https://github.com/qiaolinhan/Wildfire_Segmentation_Resnet34.

References

[1] H. Cruz *et al.*, "Machine learning and color treatment for the forest fire and smoke detection systems and algorithms, a recent literature review," in *XV Multidisciplinary International Congress on Science and Technology*, Quito, Ecuador, 2020.

[2] O. Ronneberger *et al.*, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Lima, Peru, 2015.

[3] O. Oktay *et al.*, "Attention U-net: Learning where to look for the pancreas," in *Medical Imaging with Deep Learning*, Amsterdam, Netherlands, 2018.

[4] Q.-x. Zhang *et al.*, "Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images," *Procedia engineering*, vol. 211, pp. 441–446, 2018.

[5] J. Redmon, "Darknet: Open source neural networks in C," http://pjreddie.com/darknet/, 2016.

[6] J. Redmon *et al.*, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, Las Vegas, US, 2016.

[7] F. Yuan, "A fast accumulative motion orientation model based on integral image for video smoke detection," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 925–932, 2008.

[8] F. Yuan *et al.*, "Deep smoke segmentation," *Neurocomputing*, vol. 357, pp. 248–260, 2019.

[9] X. Li *et al.*, "3D parallel fully convolutional networks for real-time video wildfire smoke detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 89–103, 2018.

[10] J. Zhang *et al.*, "ATT squeeze U-Net: A lightweight network for forest fire detection and recognition," *IEEE Access*, vol. 9, pp. 10 858–10 870, 2021.

[11] M. Jeong *et al.*, "Light-weight student LSTM for real-time wildfire smoke detection," *Sensors*, vol. 20, no. 19, p. 5508, 2020.

[12] Y. Cao *et al.*, "An attention enhanced bidirectional LSTM for early forest fire smoke recognition," *IEEE Access*, vol. 7, pp. 154 732–154 742, 2019.

[13] M. Shahid and K.-l. Hua, "Fire detection using transformer network," in *Proceedings of the 2021 International Conference on Multimedia Retrieval*, New York, US, 2021.

[14] J. Howard and S. Gugger, "Fastai: a layered api for deep learning," *Information*, vol. 11, no. 2, p. 108, 2020.

[15] J. Long *et al.*, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Boston, US, 2015.

[16] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision, (ECCV)*, Zurich, Switzerland, 2014.

[17] P. Micikevicius *et al.*, "Mixed precision training," in *International Conference on Learning Representations (ICLR)*, Vancover, Canada, 2018.

[18] S. He *et al.*, "An efficient GPU-accelerated inference engine for binary neural network on mobile phones," *Journal of Systems Architecture*, vol. 117, p. 102156, 2021.