Automated Bushfire Detection and Mapping using Image Segmentation

Anshul Jain
Department of Computer Science
The University of Auckland
Auckland, New Zealand
ajai165@aucklanduni.ac.nz

Jonathan Abiad
Department of Computer Science
The University of Auckland
Auckland, New Zealand
jabi262@aucklanduni.ac.nz

He Yu
Department of Computer Science
The University of Auckland
Auckland, New Zealand
hyu809@aucklanduni.ac.nz

Carl Chang
Department of Computer Science
The University of Auckland
Auckland, New Zealand
ccha443@aucklanduni.ac.nz

***Abstract*—Bushfires are a real world problem, and detecting them early allows response teams to act fast to save lives. Our project explored modern machine learning techniques for automated detection and mapping of bushfire boundaries on infrared linescan images. In particular, we implemented an Attention U-Net architecture for Image Segmentation due to its proven ability to work in imbalanced class distributions in other fields. For this investigation, we used the Black Summer Australian fires of 2019-2020 as our dataset. We attempted to overcome dataset shortages with an implementation of cGAN-Pix2Pix for synthetic image generation, but the method was abandoned after visual assessment and implications of using noisy synthetic images, favouring instead augmentations by geometric manipulations. We leveraged the image segmentation capabilities of Attention U-Net and performed an ablations study on the training pipeline to determine the impact of our augmentations before a final comparison with a baseline U-Net model. We have concluded that such transformations are necessary for our model, but a misclassification rate of 30% suggests that further work and improvements need to be done, especially on the dataset shortage. We also briefly touched on the architecture of Cellular Automation for future use of predicting fires within our model pipeline.**

***Index Terms*—Bushfire, U-Net, Attention U-Net, cGAN, Pix2Pix, Image segmentation, Cellular Automata**

I. Introduction

A. Motivation

Many parts of the world have been constantly ravaged by wildfires. These large-scale disasters can cause tremendous damage on the economy and the ecosystem, as well as a large loss of life. In Australia, during the summer of 2019 and 2020, rampant wildfires affected around 19 million hectares of land, resulting in over 3,000 homes burned down and 33 people dead. All this estimated to around $40 billion Australian dollars' worth of damage [1]. This period of time came known to be as Black Summer, and a retrospective analysis of wildfires [1] reveal that fires are trending to be more destructive in the future.

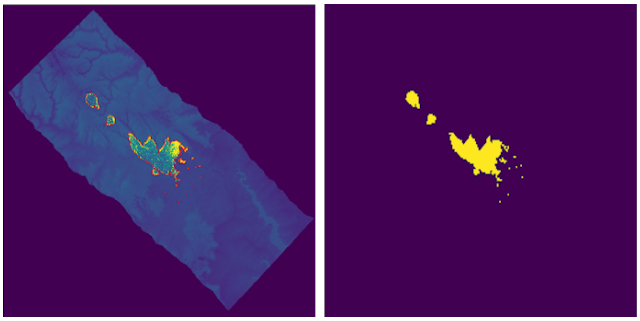

Fig. 1. From DEA (Digital Earth Australia) data set (Left) Line-scan image x. (Right) ground truth mask y.

As fires progressively worsen, the effectiveness of the fire-prevention and control methods also need to improve as the current fire mapping method are drawn by hands. For one, the combination of our satellite technology and airborne imaging can provide comprehensive coverage to determine locations of local fires quickly and efficiently. These allow us to dispatch firefighter teams and employ appropriate stopping methods before the fires grow out of control. As such, it would benefit greatly to have these systems automated. Systematic fire mapping allows for more precise and more rapidly generated data. They can provide insights on the fire's characteristics, alongside better quality of identifying areas susceptible to new fire starts and spreads [2]. With modern methods, such systematic approaches are possible. Many studies have already been conducted on using machine learning and deep learning algorithms to automatically extract features from images, many of which with application to fires [3] [4]. Essentially, with a more automatic and advanced method of fire mapping, analysis and spread would increase our effectiveness in reducing damages caused by fires.

B. Problem Statement

With the direction of building an automatic system to detect fire areas settled, we explored that based on the observations of infrared linescan images as shown in Fig. 1, which method could be helpful to detect fire areas precisely. The inspiration to solve this problem is if the images can be considered a combination of each pixel. Within each small region, the combined information of each pixel and its surroundings will be given the probability of fire. With further literature survey, Image Segmentation becomes the most effective method to solve this problem.

This investigation will examine how we can use and improve Image Segmentation via the U-NET architecture to construct an automated approach for fire detection and analysis. Specifically, Attention U-NET has already been proven to be effective for studying neuron structures and pancreatic analysis [3]. This model is designed for classifying small regions of interest in images, and applying this in a bushfire application can increase our accuracy when searching for small-scale fires that need to be put down early. Thus, our objective is to construct an architecture using Attention U-NET and compare it with existing U-NET fire methods to forecast fire-regions given the provided images. The regime of fire mapping can be defined as an image binary classification problem, as we are deciding where in a given image does fire exist, and where it does not.

Furthermore, we will also analyse the next steps to take once our image mapping has generated the desired output. This mainly involves fire prediction, which delves into extrapolating fire behaviour to ensure full coverage about a given fire whenever imagery for our model via airborne or satellite detection is not available. Moreover, taking these steps can enable us with thorough information regarding a fire's potential path and give us new insight about its behaviour.

C. Contribution

Our work focuses on applying a medical image segmentation technique to the bushfire problem and identify the usefulness of the approach. As this is a new application of the model, we will need to review the method and determine if the assumptions made by previous models are applicable for the development for ours, then adjust as necessary. Neural Networks, particularly ones designed for Image Recognition of bushfires have a problem with identifying small images, and improvements could be made further to address these problems [5]. Although there are many approaches to address this problem, we have chosen U-Net as it has been proven effective for identifying minority classes in the medical field via down-sampling then recombining for up-sampling. This will be discussed further in our Related Works section.

In addition, there are three common solutions to solve imbalanced classes problem, which includes adding attention mechanisms to the network structure, choosing suitable loss functions to focus small object and up-sampling the minority class. The first two methods was introduced to our project and the details will also be discussed in the next section. We have also endeavoured to expand on other areas and investigate other algorithms that could aid us in several problems identified within this investigation. This includes the usage of cGANs to fix dataset shortage problems and investigation of cellular automata simulations for fire prediction. Although not completed, these steps have given us potential pathways for the improvement and expansion of our work in the future.

II. Related Work

A. Image Segmentation

Image segmentation is a thoroughly covered image processing technique used throughout many disciplines. This approach partitions regions of pixels in an image into specific class labels representative of the whole region. Semantic segmentation is a classic version of this, where the specific algorithm would focus on a certain image, also known as a semantic object, and the appropriate segmentation to partition that object against all other classes. For instance, within the medical field, image segmentation is used in action recognition systems, where different body parts are identified and assigned a class [3]. With the advent of neural networks and more advanced machine learning, the accuracy and efficiency of automated image analyses have only continued to improve. An investigation of deep learning techniques for Image Segmentation [3] provided a thorough list of approaches we could have potentially used for our own project.

B. Convolutional Neural Networks

Before the onset of deep learning strategies, image segmentation was traditionally done based on clustering algorithms with histogram thresholding. These require more parameters for contour and edge inputs, and part of the process is done manually. Neural networks have revolutionized this approach, especially with the introduction of Convolutional Neural Networks (CNN). What sets apart CNNs from a Deep Learning Approach is the convolutional layers, where filters can easily process the image through convolution without the added complexity of adding extra neurons or layers. Pooling layers also exist in the CNN, which serves to reduce the image size by taking representative values of a region. The final layers convert the 2D layer into a linear output similar to that found in traditional neural networks which are then processed as a new segmentation map with class values.

C. U-Net and Attention U-Net

1) U-NET: U-Net has also been termed "The 100 layer Tiramisu"[6] and was introduced to make significant improvements in the field of biomedical image segmentation. A winner of cellular tracking challenge, this architecture is an encoder-decoder structure with the added advantage of skip connections. These connections are applied to improve on the abstraction problem in auto-encoder models. Skip

connections simply combine the outputs of different depths between the encoding and decoding phases and that has shown to be very effective in producing low noise output masks. U-Net has proven to be a solution in a category of image segmentation problems where training images are few and reliance on data augmentation is strong.

2) Attention U-Net: Attention U-Net was introduced to tackle another medical image segmentation issue of identifying pancreas cells (target) of varying size, which is a minority compared to other cells in the image. Attention gates were introduced in the paper, that allowed the model to focus on target (salient) region of the image, decided to be important by the model, while suppressing all other regions. The feature maps created during up-scaling and by skip connections, are passed through attention gates, that help determine the importance of each feature. This mechanism helps reduce the effect of unnecessary regions of the mages and focus more on the important part of the image related to the prediction task.[6] The limitation of Attention U-Net is the additional computational overhead by the time it takes to compute weights per pixel and garner attention parameter. The proposed model to solve our problem statement will be U-Net and Attention U-Net. The architecture is comprehensive and is suitable for 2D images, and can be amended to train on 3D images when spatial information is provided as a form of input.

D. Potential Loss Functions

With the research to solve image segmentation on imbalanced data, there are three loss functions used in our project: Binary Cross-Entropy (BCE) [7], Weighted Binary Cross-Entropy (WCE) [8], Dice Loss [8] and the combination of Dice Loss and Binary Cross Entropy (DiceBCE). The advantages of using these four loss functions are that no parameter tuning is required. Dice Loss focus on the similarity of the overlapping area between the prediction and the ground truth, which is defined based on the evaluation metric:

$$L_{DL} = 1 - \frac{2|A \cap B|}{|A| + |B|} \tag{1}$$

where A and B are used to represent the segmented mask for the ground truth and the predicted images.

The use of weighted binary cross-entropy focuses on the difference of the probability distribution between the ground truth and the predicted masks, which is defined as:

$$L_{WCE} = \frac{1}{N}\sum_{i=1}^{N} -[wy_i \cdot log(p_i) + (1-y_i) \cdot log(1-p_i)] \tag{2}$$

where y_i stands for the true label of instance x_i, 1 for positive and 0 for negative, p_i stands for the probability of predicting the positive, w stands for the weight and needs to be calculated based on the overall probability of the positive class in the training-set, $w = 1$ when the positive class weights the same as the negative class. DiceBCE can be considered the summation of Dice Loss and BCE loss.

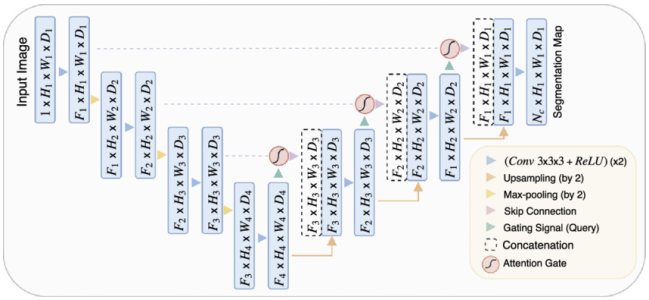

Fig. 2. Attention U-Net architecture [11]

E. Fire Prediction and Cellular Automation

For this project, we have also studied various fire simulation methods for bushfire prediction. We mainly considered cellular automata models due to its ability to factor in many predictors whilst being simplistic in its construction. Cellular Automata consists of cells, which in our case would be the pixels within the image. These are governed by a set of rules which control the state of each pixel. On an individual level, these may seem like random behaviour, but the pixels demonstrate emergent behaviour as a whole. Many fire simulations have been modelled using cellular automation [9] and one study uses a model implementing Rothermel's equations of fire spread [10] for testing resulting masks from our Fire Mapping models.

III. Dataset and Features

We have utilized satellite images and airborne infrared linescan images provided by Digital Earth Australia (DEA) and NASA for the creation of our model. This is conveniently stored via a Geographic Information System (GIS), which administers metadata of our fire images such as time and coordinates, transformations applied to flatten the earth's topography into a 2D map as well as vectors of various polygons that make up features of the landscape. Part of the challenge was understanding how to utilize and extract information from a comprehensive information system before pre-processing the data. This was mostly overcome, though there are still gaps within our understanding of GIS data.

The data provides 7 different fire events during the Australian Black Summer fire, as well as an 'Other' category. These have been split into around 130 training images and 5 unlabelled images for testing. These are called linescan images, and are taken using infrared imaging via airborne and satellite means. Polygons denoting fire had been manually drawn on top of each linescan image using conventional means. Each image provides a different component of the area that had fire, although only 37 of these images had polygons corresponding to a fire mask. Attention U-NET requires many observations to create well-fitted models, and we have decided to approach this shortcoming using various data augmentation techniques, which will be further discussed in the pre-processing step. Similarly, all images come in different dimensions which have to be accounted

for in our preprocessing step, even though U-Net has been able to handle inputs of different sizes in past architectures.

IV. Methodology[1]

A. Augmentation attempt with cGAN-Pix2Pix

Initially we decided to boost the dataset set by synthetically generating images using training a generator model. We selected the use an implementation of conditional Generative Adversarial Network (cGAN) referred to as Pix2Pix [12]. The implementation allows image to image translation, where the input image is the ground truth mask, and the output is a synthetically generated linescan image. Both the generator and the discriminator are trained at the same time, where the generator creates synthetic images, the discriminator tries to distinguish between real and synthetic images, then weights are updated through backpropagation according to the objective function, where x is the linescan image, y is the ground-truth mask, z is the initial input noise, D is the discriminator, G is the generator and λ is a tune-able parameter where [12] described it to have the best results when set to 100:

$$\begin{aligned} G^* = \arg\min_G \max_D \mathbb{E}_{x\sim p_{data}(x)}[\log D(x|y)] \\ + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y)))] \\ + \lambda \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1] \end{aligned} \tag{3}$$

Our objective is for the generator to learn to generalize over the distribution of the linescan images x over input noise z given the linescan images y. The data loader first standardizes the input size of each image by re-scaling them to size 256x256, then the cGAN was trained over 2500 epochs using 32 image-mask pairs, and 5 image-mask pairs for validation.

After training, we took some of the original masks and applied random geometric transformations (horizontal-flip, vertical-flip, rotation, resize) to them, then using these "new" masks as input to the generator model. We visually assessed the resulting generated images Fig.3 and found artifacts (Red patches), noise and low-quality.

The team decided that it would not be sensible to move forward with adding low-quality, noisy, artifact infested, images to the training data set. Due to time constraint, and the complexity of diagnosing the removal of image artifacts, the team decided to abandon the idea of using cGAN generated images for augmenting our dataset.

B. Preprocessing

1) Converting to grayscale: As we are only interested in detecting the fire boundary, the fire and non-fire areas indicated by the red polygons is well-contrasted in grayscale. By removing the colours, this reduces the complexity of features that our U-Net model will need to learn, ultimately reducing the training time and resulting in a significantly less complex model.

[1]https://github.com/FoundingTitan/Fire_Mapping_Project

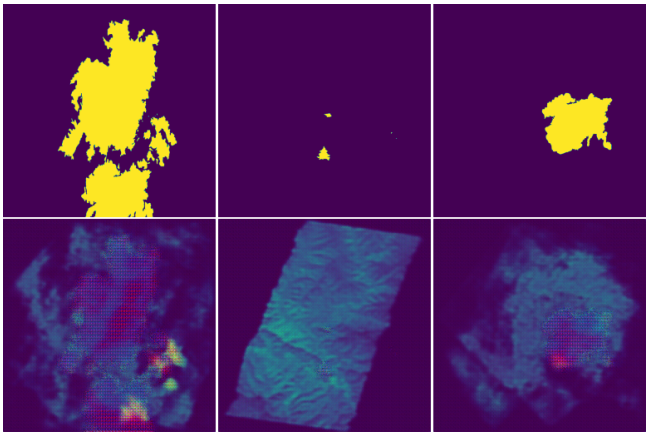

Fig. 3. cGAN generated linescan images. (Top-row) Input mask (y), (Bottom-row) Synthetically generated linescan images ($\hat{x}$). The left and right images show unwanted artifacts as red patches.

2) Resizing: The training images are of various sizes, and need to be standardized such that all images are of the same size; since the data batch loader requires all images of one size. We chose to re-scale them to 256x256 because the majority of each of the images is mostly zeroes, choosing any larger scales would be computationally wasteful when filters convolve over the images mostly comprising of zeroes.

3) Data augmentation: To generate a larger dataset without synthetic generation, we decided to geometrically transform all 30 image-mask pairs in our training set. This is done by flipping all original image-mask pairs once, both vertically and horizontally. The final training set consists of 30 image-mask original pairs, 30 vertical flipped image-mask pairs and 30 horizontal flipped image-mask pairs, totalling a 90 image dataset.

C. Ablation study

The ablation study is on the machine learning pipeline. In the following section we will refer to geometric transformations (Random crop, vertical flip, horizontal flip) on the image and mask pair as an augmentation component. To prove the effectiveness of our models, the performance of the model trained with full augmentation is compared with various models trained with some augmentation component removed (See Fig. 6) along with the baseline model (U-Net trained on 30 image-mask pair).

D. Model Building

The U-Net model inspiration was taken from the original paper [13] to implement five 2-D convolution layers, each in the downward and upward direction. Every block passes the input into a batch normalization function which is then passed through each kernel, after which weights are applied, the output is passed to a 2-D max-pooling of size two and stride of two, followed by a ReLU function. The 2-D kernel in each layer is of size three, stride of one and padding of one. The output is passed through a final normalization

function before leave that convolution layer. The number of kernel doubles in each layer. The first convolution layer applies 64 kernels, the second layer applies 128 kernels, followed by 256, 512 and 1024 kernels respectively. Skip connections are applied in between layers in each direction. The output from each downward direction (down sampling) layer is copied then cropped and finally concatenated with the corresponding upward direction layer (up sampling).

The Attention U-Net model utilizes three attention gates to extract hard and soft region features and highlight the minority class in an image. Each attention gate is a block of 2-D convolution later of kernel size one, stride of one and no padding. The resulting mask passes through a batch normalization within the attention block and then passes to a ReLU activation function. The number of up-sampling and down-sampling convolution blocks are kept the same as the U-Net model.

The size of every input images was different from each other in the data set. The model resized the images to 256x256 and converted them into tensor before allocating into batches. The motivation to convert the input into relatively small size of 256x256 was because the proportion of the class of interest (fire pixels) are significantly smaller than non-fire area (dark pixels) in every input image. A larger size of 512x512 would mean the filter has to perform computation on dark pixels proportionately more, but will not learn anything from them, and result in computationally and time expensive epochs. The batch size was calculated by data loader during enumeration and was dependant on the number of images used in training. For example the original data set of 30 images was divided into a batch size of 4 images going into training in 8 batches per epoch. The model was designed to invoke multiple transformations of input images before training, as a value proposition to test effectiveness of individual transformations. The transformations implemented were to either crop image to size of either 256x256, 280x280 or 300x300. A random generator was used to determine if the image should be cropped or not, and another random generator to select which of the three sizes should be selected to crop the image. The second and third transformation options were to flip the image on their horizontal and vertical axis. The three transformations were selective, such that the user, while training has control over which (or all) transformation should be applied to training data. All transformations were encoded to be enforced by a random generator, to remain unbiased.

E. Hyperparameters

The hyperparameters give a model the flexibility of training on the dataset, i.e., they are often used for helping estimate the parameter of the model. They are often set before training according to experience. In our model, mainly three hyperparameters can be tuned: learning rate, number of epochs, and batch size.

1) Learning Rate: The learning rate represents how fast the model can learn; the higher the rate, the more accessible

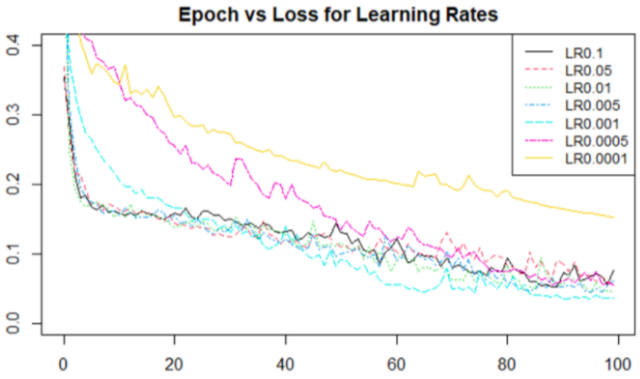

Fig. 4. Different Learning Rates on Training Loss

learning, which also implies a more difficulty in converging to a best solution. The learning rate is tuned by the babysitting method, where we set a series of learning rates that are valid from 0.0001 to 0.1; in total, there are seven rates to be selected. After the training for each learning rate, the best learning rate is selected based on the lowest training loss. As shown in the Fig. 4, the best learning rate to lower the training loss is 0.001. Empirically, this learning rate is not high to avoid converging, and the training time is reasonable.

2) Epochs: The number of epochs can be considered how many iterations the model requires for training a proper model without underfitting or overfitting. The common method is to set an arbitrarily large number of epochs while in the first training and observe which epoch the training starts to converge while the validation loss starts to stop decreasing. This method is as early stopping and is used frequently in other neural network architectures. [14]. Theoretically, the early stopping can happen in any training progress paralleled, so we decide to observe it based on the Fig. 5 which will be talked in the next section; in most of the cases, the training loss starts to converge after 100 epochs, where the validation loss starts to increase. Hence, according to early stopping, we decide to train up to 100 epochs only.

3) Batch Size: In the last, we tend to tune the batch size. To tune the batch size, first, we should understand what does batch size do. The batch size decides the direction of gradient descending [15], for example, if the batch size can process the whole dataset (full batch learning) and make the decision that this direction is the best direction for the whole dataset, the model will return the prediction based on this direction. However, when the dataset is too big, and the data is too unique, which is the same situation as we have, the full batch learning may ignore the difference of sampling in the dataset and consider the overall performance. Therefore the correction of gradient descending counteract each other and could not give a proper direction. The method to choose a proper batch size is called Mini-batches learning [16], which reduce to half batch size while training, such as 16, 8, 4, 1. Notably, for our data, the 16 batch size returns low precision initially; to reach the higher precision, the cost of time

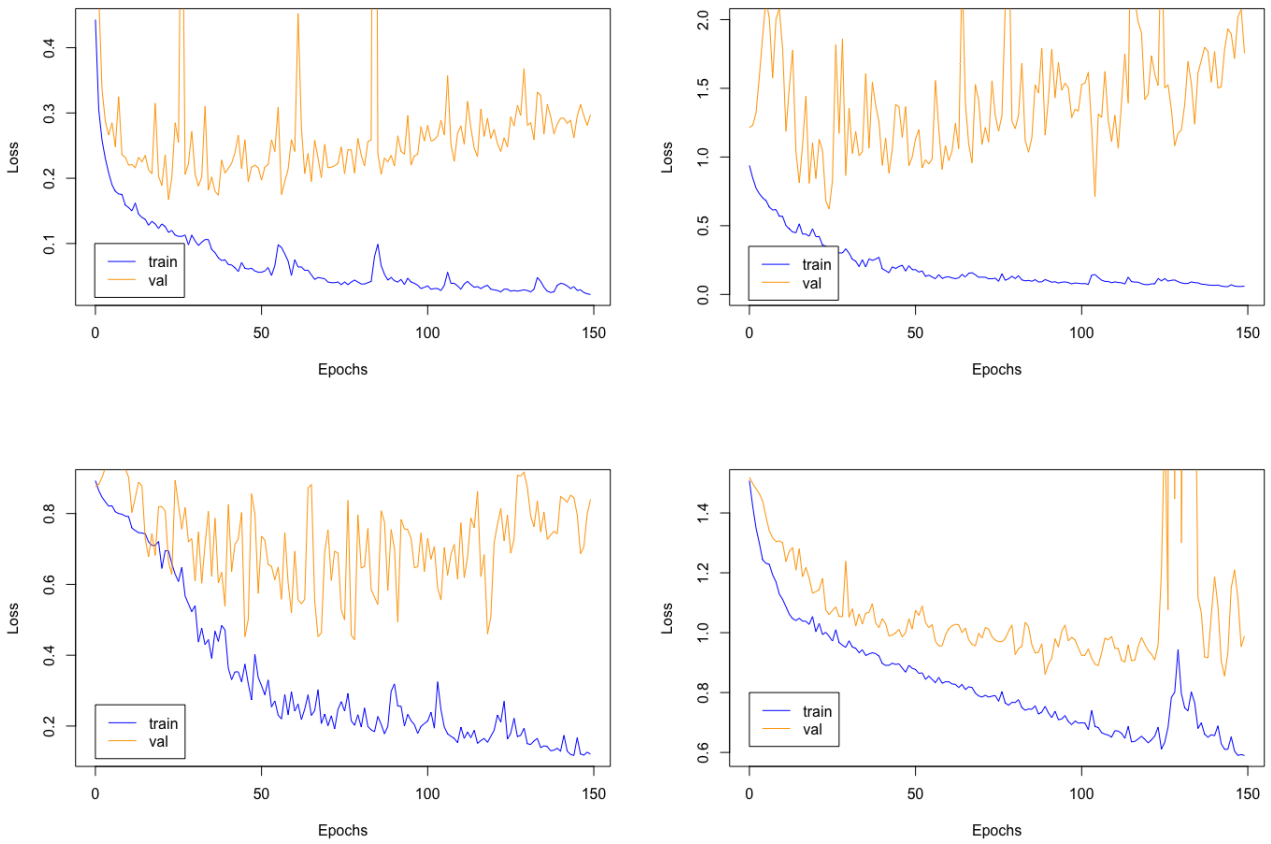

Fig. 5. Training vs. Validation Loss on BCE, WCE, Dice Loss and DiceBCE Loss (top left to bottom right)

increases substantially, and the direction remains unchanged. The model does not converge when choosing batch size as one (online learning). Due to the tradeoff between full batch learning and online learning, the experiments show that the suitable batch size is 8 for the model.

F. Loss Functions

As mentioned in the related work, four loss functions will be implemented in our model and compared based on their performance. The performance metric is mainly training loss and validation loss, which indicate the fitness and the explanation of the model, the other performance metric is how the overall F1 score performed.

Firstly, the output of the model is fitted into the Sigmoid function to normalize the probability matrix. Since the data only include two classes, after the normalization, when the probability is higher than 0.5, the result should be considered as a positive class/fire. The binary cross-entropy can be calculated by using Formula 1. While the loss functions are implemented and used to train and validate the model, the two loss for the training and validation set is presented as the loss items during training and validating. If both of them can be converged and share similar loss items, the model is valid for fitting the model; otherwise, if the training loss is going down alone while the validation stays the same or cannot converge, the model is considered as overfitting. The Figure. 5 shows how loss is calculated for each loss function.

As shown in the figure, only BCE Loss and DiceBCE Loss tends to converge at the end of training. Both WCE Loss and Dice Loss tends to overfit the model, and the validation curves are unstable to explain if the model is good to be used. WCE Loss fits the training set well, however the distribution for the validation set is unknown and could be completely different as the training set, hence WCE is regarded as invalid. Moreover, even dice loss tends to improve the overall F-measure to 0.7, but the increase is not statistically significant to ignore the terrible loss. In addition, the reason why dice loss can not be converged is because the gradient format for dice loss is $\frac{2t^2}{(p+t)^2}$, where t presents the target and p presents the prediction. It is clear that when both prediction and target are small, if the prediction went wrong, the dice loss will increase dramatically and the training becomes unaccomplished. Therefore, the choice is limited into BCE Loss and DiceBCE Loss. Both of them converged, but the volatility appears in the end of training which follow the same principle of Dice Loss, the explanation for choosing

DiceBCE is feeble, especially when the model is difficult to be trained and the loss cannot decrease as expect. Hence, the choice of loss functions stick into the Binary Cross-Entropy.

G. Evaluation Metrics

Our original plan is while the model can be converged to a certain value (less than 0.5) on both training loss and validation loss, the best model will be chosen by the best F1-measure on the certain epoch, which indicates the balance of precision and recall. However, we have recognized if the dataset is highly imbalanced, the F1-measure may not be enough to be considered the only performance metric. Instead of finding the harmonic mean of precision and recall, the model is supposed to focus more on the recall than precision. Therefore, to pick the epoch and its resulting model, we have evaluated each model on the validation set with all possible cut-offs. This allowed us to calculate the Area Under the Curve (AUC) of each model based on Precision Recall (PR) Curve, which became the metric by which we selected the best model for all runs. PR curve is the standard way to observe the tradeoff between recall and precision, especially when the prediction favours one of them rather than both of them. Care must be taken to select a model that is well-fitted, and not underfit or overfit. Thus, our epoch selections involved preferring epochs halfway through the training session, and not just optimizing for the best AUC.

Furthermore, as the model will output a probability matrix, it was necessary to decide on the best cut-off to determine a model's final prediction of fire regime within an image. To determine the optimal cut-off, we had a U-Net architecture generate the 100 epochs, selecting the best AUC and tested on the evaluation set to determine the resulting Precision-Recall Curve. We then optimized for the F3-measure based on each pair of precision and recall, which differs from F1-measure by the way it provides more weight for the Recall statistic. The way how weight applies to recall or precision is based on the F-measure:

$$F = \frac{(1+\beta^2) \times Precision \times Recall}{\beta \times Precision + Recall} \tag{4}$$

This prevented our cut-off from being too high and still achieving a high precision due to most pixels being naturally absent of fire. This U-Net model, which will be discussed further in the next section, yielded a best F3-measure at a cut-off of 0.52. This cut-off has been used for the generation of our U-Net and Attention U-Net model results.

V. Results

A. U-Net and Attention U-Net Outputs

For the first part of our testing, we have applied the original U-Net architecture to our bushfire dataset. This presents us with a baseline in which we can compare the performance of future Attention U-Net models. The output of the resulting mask for our test set is shown in Fig. 6. As presented in Table I, U-Net appears to have taken a more balanced approach in our dataset, with near-equal precision and sensitivity parameters. The resulting images suggest that it appeared to struggle with smaller fires and more specific areas. The boundary and size of the fire was overestimated at times as well, and judging by the numbers, it performed inadequately with the amount of data provided. Its precision is around 0.527, just a slight improvement above random guessing.

On the other hand, with the same conditions, an Attention U-Net model over-predicted the size of a fire, more so than the baseline U-Net model. The sensitivity rate of the model improved from the baseline drastically, up to around 0.937, but at the cost of a much lower precision. This is near-equivalent to predicting everything as a fire zone. Although a high sensitivity is desirable, these sorts of predictions are unusable in a practical sense. To improve on this, we add our 60 augmented images and train again on the Attention U-Net model. This improved the precision of the model immensely, but now the reverse is true as with the original Attention U-Net model where now the sensitivity is too low at 0.187. This could potentially be due to the fact that the 90 images in the augmented dataset are similar to one another, and the increase in precision but decrease in sensitivity could be a result of overfit, or simply our model predicting most things as 'not fire', as suggested by the images in Fig. 6.

Next, we transformed the expanded dataset of 90 images with further transformations, random crop, vertical flip, horizontal flip and a combination of all three. Each additional transformation on its own improved the sensitivity of our results. This improvement is minimal, as our sensitivity only increased from 0.187 to 0.326. This suggests that individual random transformations vary our images, especially the augmented images, which is necessary due to the lack in the original data. The combination of all three transformations resulted in our best result throughout this investigation, with a sensitivity more comparable to the original baseline U-Net model while at the same time having a much greater precision at 0.714. Thus, the added transformations vary enough for our model to predict our test set with decent performance metrics.

B. Fire Prediction

Our GIS Dataset also contained information for different bandwidth filters to extract different filtered images from our dataset. This would give us the necessary vegetation parameters to add to our cellular automation model. We have gone as far as to replicate simulations with zero initial conditions apart from the original fire locations provided by either our model or from the dataset. This produced simulations similar to those provided by prior research [10] [17], although without additional parameters, travelled in all directions, compared to the mainly singular direction provided in the real dataset. However, we did not fully understand the systems behind different filtered images and

TABLE I
RESULTS OF VARIOUS TESTS ON U-NET AND ATTENTION U-NET MODELS

Model	Transform	Precision	Recall	Specificity	NPV	AUC
U-Net	None (Baseline)	0.527	0.685	0.920	0.970	0.621
Attn U-Net	None	0.241	0.937	0.821	0.980	0.649
Attn U-Net	Expanded dataset	0.838	0.187	0.998	0.943	0.655
Attn U-Net	Random Crop	0.710	0.216	0.998	0.948	0.700
Attn U-Net	Vertical Flip	0.745	0.277	0.995	0.949	0.632
Attn U-Net	Horizontal Flip	0.720	0.326	0.996	0.949	0.707
Attn U-Net	All	0.714	0.525	0.989	0.959	0.667

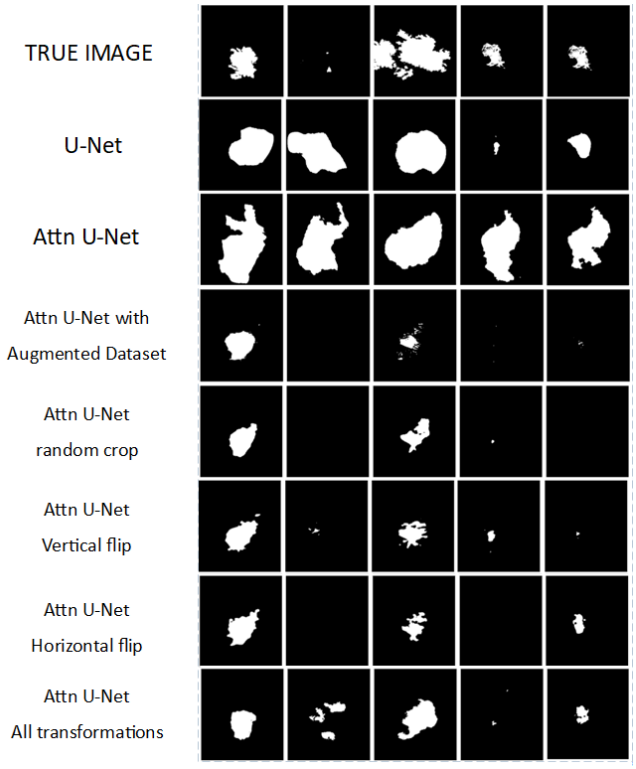

Fig. 6. Predicted Masks from 5 Test Images

how to extract or utilize them for this application at this current time. Therefore, we have decided that a Cellular Automata model for fire spread prediction would be more beneficial for future work as an extension to the fire mapping model.

VI. CONCLUSION

To conclude, we have created a framework for the automatic generation of fire zones given provided airborne linescan images using the Attention U-Net architecture. This was an improvement over the traditional U-Net model typically used in other applications. We have compared the performance of different components of the pipeline, including various augmentation steps, and determined that all steps are necessary in order to generate the best model. Essentially, our method transformed the original U-Net model from generating high sensitivity masks to one that generated high precision This was followed by a gradual improvement of the sensitivity until it is similar to the original U-Net model again, while maintaining a decent precision score. In this regard, we can deem our improvements to be successful. However, the resulting images are still not usable for commercial application, and many more improvements can be done to improve on our results.

A. Discussion

As shown in the result section, with more components added to the U-Net model, the model performance improves with around 0.7 for precision and 0.5 for recall. After a comprehensive study of various combinations of augmentations and hyperparameter tuning of our model through ablation, the feature of the attention mechanism is revealing. For example, when the training set is only limited to 30 images, Attention U-Net even give a much worse result. The attention mechanism failed to improve the performance because when there is not enough data provided, attention tends to determine the importance of each class recklessly. Hence the probability of positive can be easily overestimated as well as results in the higher sensitivity.

The realization of a lack of data, especially for Attention U-Net, is critical, leading the model to the next step, an Expanded Dataset. The implementation of the expanded dataset is booming, which improves the precision of the model. Moreover, the data augmentation method is also applied randomly to the expanded dataset while training to improve the recognition of the dataset on attention mechanism. However, the result is always unfavourable when the model tries to predict the small fire area like the second test image. To explore why the training set and test set get reviewed, we explore the potential limitations of our model.

B. Limitations

One of the major limitations for the project comes from the dataset. As our project uses a real-world dataset, the

number of training examples is small, as not all linescan images were provided with a ground truth image mask. Some of the test images do not have proper and complete polygons that exactly outline the fire boundaries, and could not be used for training. Furthermore, when attempting to boost the size of our dataset through the use of cGANs, we could not overcome the generation of artifacts, and resolution quality issues. Furthermore, our solution only explored image segmentation of detecting image fire boundaries in greyscale, and have not explored fire detection through coloured linescan images, and do not know if using coloured images would improved our current results.

C. Future Work

It is paramount to improve on the model's performance and accuracy to find success in the industry as a useable fire mapping tool. As our models have an approximately 30% misclassification rate, this is still potentially achievable given the problems encountered within this investigation. The most predominant of these is the lack of training data. Our dataset only contained 37 linescan images, and as a CNN that needs hundreds of images for proper training [18], our Attention U-Net model could potentially still be underfit despite all the additional augmented data we provided. The fact that it managed such statistics and performance metrics with a low amount of training images means that there is good potential in the model, and our accuracy rate could improve further. Additionally, once we have the extra data, we would need to tune more hyperparameters. This includes, and is not limited to epoch training time and batch sizes, and a re-investigation of the appropriate learning rates and optimal cut-offs. Even with a larger dataset, we still run the risk of overfitting, and the balance between a well-fit model and training efficiency is all the more important with a larger dataset, as epochs can potentially run at a much longer duration. Moreover, the output masks need to be transformed back to its original size and super-positioned onto the required image for a particular fire to be mapped. This is an area we would definitely need to address in a future study, should we continue to use the models that we have implemented.

Additional improvements to our model performance could also include factoring in distance of false positive and false negative predictions. For example, a distant false positive is more problematic than a false negative that is part of the same fire segment as a correctly predicted fire as there would be a waste of precious time to deploy and redeploy firefighting teams to stop fake fires. Once we have a fully working model, it is also important to realize what these predictions mean and how it would impact the application at hand, and addressing distances is a step in this direction. We could attempt to do this by firstly investigating models that factor distance into their loss functions, then examining the output probability matrix just before the cut-off step. We can then have an adaptive cut-off that grows based on a cell's distance to high probability fire zones. This is just one example of a possible approach, and further investigation could reveal more effective and efficient solutions to this problem.

Further work could also extend to the regime of fire prediction. This was a topic that was initially planned for the model pipeline and attempts had been made to create a working simulation. However, plans for a fire prediction model saw a lack of progress due to time constraints and need for further understanding of the parameters used within the image. Nonetheless, this is a field with a very large potential for our Australian Bushfire Dataset as many fire simulations have already been generated using this approach [9]. Such a model can work in conjunction with our fire mapping model, and further development in all these areas could lead to a fully automated and integrated fire detection and mapping pipeline for industry use in the future.

Authors' Contributions

AJ investigated and proposed U-Net and Attention U-Net for the problem statement. Accumulated the training images and converted them into masks. Also wrote the complete training, testing and validation code. JA investigated cellular automation, ran tests with AJ and HY, wrote evaluation and basic augmentation code and was the lead editor of this manuscript. HY investigated how loss functions and hyperparameters will influence the model performance especially towards imbalanced data and also the limitation and discussion. CC investigated the use of GANs and its variants for data augmentation, along with the code for implementing it and writing README.md instructions for running the code. All authors reviewed the final manuscript.

Acknowledgment

The authors would like to thank Katerina Taskova for providing the idea of ablation study and valid evaluation methods for the model. We would like to thank Kaiqi Zhao and Katharina Dost for mentioning the optimization for imbalanced data based on the loss functions.

References

[1] A. I. Filkov, T. Ngo, S. Matthews, S. Telfer, and T. D. Penman, "Impact of australia's catastrophic 2019/20 bushfire season on communities and environment. retrospective analysis and current trends," *Journal of Safety Science and Resilience*, vol. 1, no. 1, pp. 44–56, 2020.

[2] S. C. Avitabile, K. E. Callister, L. T. Kelly, A. Haslem, L. Fraser, D. G. Nimmo, S. J. Watson, S. A. Kenny, R. S. Taylor, L. M. Spence-Bailey *et al.*, "Systematic fire mapping is critical for fire ecology, planning and management: A case study in the semi-arid murray mallee, south-eastern australia," *Landscape and Urban Planning*, vol. 117, pp. 81–91, 2013.

[3] S. Ghosh, N. Das, I. Das, and U. Maulik, "Understanding deep learning techniques for image segmentation,"

ACM Computing Surveys (CSUR), vol. 52, no. 4, pp. 1–35, 2019.

[4] P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," *Case Studies in Thermal Engineering*, vol. 19, p. 100625, 2020.

[5] J. Zhang, H. Zhu, P. Wang, and X. Ling, "Att squeeze u-net: A lightweight network for forest fire detection and recognition," *IEEE Access*, vol. 9, pp. 10 858–10 870, 2021.

[6] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 11–19.

[7] M. Yi-de, L. Qing, and Q. Zhi-Bai, "Automated image segmentation using improved pcnn model based on cross-entropy," in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.* IEEE, 2004, pp. 743–746.

[8] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. 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*. Springer, 2017, pp. 240–248.

[9] W. Velásquez, A. Munoz-Arcentales, T. M. Bohnert, and J. Salvachúa, "Wildfire propagation simulation tool using cellular automata and gis," in *2019 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2019, pp. 1–7.

[10] A. Alexandridis, D. Vakalis, C. I. Siettos, and G. V. Bafas, "A cellular automata model for forest fire spread prediction: The case of the wildfire that swept through spetses island in 1990," *Applied Mathematics and Computation*, vol. 204, no. 1, pp. 191–201, 2008.

[11] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[14] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[15] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, "Don't decay the learning rate, increase the batch size," *arXiv preprint arXiv:1711.00489*, 2017.

[16] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, p. 2, 2012.

[17] X. Li, "Wildfire simulation using paralleled cellular automaton," 2018. [Online]. Available: https://github.com/XC-Li/Parallel_CellularAutomaton_Wildfire

[18] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.