Pixel-Wise Machine Learning and Deep Learning Methods Implementation on Multi-Class Wildfire Mapping

Mingda Wu
Northern Illinois University, Z1940079@students.niu.edu

Follow this and additional works at: https://huskiecommons.lib.niu.edu/studentengagement-honorscapstones

Part of the Environmental Monitoring Commons

Recommended Citation
https://huskiecommons.lib.niu.edu/studentengagement-honorscapstones/1464
NORTHERN ILLINOIS UNIVERSITY

Pixel-Wise Machine Learning and Deep Learning Methods Implementation on Multi-Class Wildfire Mapping

A Capstone Submitted to the

University Honors Program

In Partial Fulfillment of the

Requirements of the Baccalaureate Degree

With Honors

Department Of

Earth, Atmosphere and Environment

By

Mingda Wu

DeKalb, Illinois

May 13, 2023
University Honors Program
Capstone Faculty Approval Page

Capstone Title (print or type)
Pixel-Wise Machine Learning and Deep Learning Methods Implementation on Multi-Class Wildfire Mapping

Student Name (print or type) Mingda Wu

Faculty Supervisor (print or type) Wei Luo

Faculty Approval Signature

Department of (print or type) Earth, Atmosphere and Environment

Date of Approval (print or type) 5/2/2023

Date and Venue of Presentation April 18th, 2023, at CURE in Holmes Student Center

Check if any of the following apply, and please tell us where and how it was published:

☐ Capstone has been published (Journal/Outlet):

☐ Capstone has been submitted for publication (Journal/Outlet):

Completed Honors Capstone projects are used for student reference purposes electronically in Huskie Commons.

If you would like to opt out and not have this student’s completed capstone uploaded to Huskie Commons for reference purposes, please initial here: _______ (Faculty Supervisor)
Abstract

Wildfires are destructive natural hazards. Artificial Intelligence (AI) has been a trendy topic in recent years due to its powerful applicability. This study focuses on the use of artificial intelligence (AI) in hazard management, specifically in the field of wildfire mapping. Machine learning and Deep learning are two subsets of AI. This study applied pixel-wise machine learning and deep learning methods to do multi-class mapping on two wildfire events in California, USA. The purpose of this research is to demonstrate the usefulness and advantages of using AI in the field of hazard management. The machine learning methods selected are Random Forest, eXtreme Gradient Boosting and Support Vector Machine. The deep learning method used is U-Net. The results indicate that U-Net did the best job at classifying wildfire events, while SVM had the best performance among machine learning algorithms. U-Net is, however, the most time-consuming model due to the nature of deep learning. There are some aspects of this study that can be improved. The models may be tuned to have better performances. And it will be better to use hand-labeled masks to make the deep learning model more useful in more complex conditions. This research emphasizes that the use of AI in hazard management can improve the accuracy and efficiency of wildfire mapping. It also highlights the potential for AI to be applied in other fields of hazard management. Overall, the study demonstrates the usefulness and advantages of using AI in wildfire mapping and provides insights into how this technology can be further optimized for hazard management.

Keywords: Wildfires, Multiclass classification, Multiclass semantic segmentation, AI, Machine learning, Deep learning, U-Net, Pixel-wise, California, Remote sensing
1. Introduction and Background

Wildfires are destructive natural hazards that can cause great damage to the environment and human society, and they are becoming more frequent and intense due to climate change, and California is one of the most affected regions. These natural disasters pose a significant threat to the environment, infrastructure, and human life. In recent years, Artificial Intelligence (AI) has become a popular tool to tackle complex problems in various fields, including disaster management. Machine learning, a subset of AI, has shown promise in addressing the challenges of wildfire management. Machine learning requires some data to train the model and then apply the model on the new data to make predictions. Deep learning, a subset of machine learning, uses neural networks to increase the performance of the model. These approaches involve training algorithms to recognize patterns in data, which can help predict the likelihood and severity of a wildfire, as well as aid in mapping the affected areas. In this study, the focus is on using pixel-wise machine learning and deep learning methods for multi-class mapping of two wildfire events in California, USA. The three machine learning methods selected for this study are Random Forest (RF), eXtreme Gradient Boosting (XGBoost) and Support Vector Machine (SVM). The deep learning model selected is U-Net (Ronneberger et al., 2015). Their results will be compared using the same accuracy metrics. These algorithms have been widely used in the field of remote sensing, and they are particularly effective in classifying high-dimensional data with complex relationships. The study will compare the accuracy metrics of the four methods to demonstrate the usefulness and advantages of machine learning and deep learning in the field of hazard management. The benefits of AI in wildfire management are numerous. AI can help identify high-risk areas, which can inform decision-making about resource allocation and evacuation procedures. It also can aid in post-disaster recovery efforts by mapping the affected areas and
assessing the extent of damage. This information can assist in prioritizing restoration efforts, identifying areas that require immediate attention, and estimating the cost of recovery.

2. Dataset and Methodology

The data for this study was satellite images from Sentinel 2 MSI Level 2A product (Sentinel-2 - Missions - Sentinel Online - Sentinel Online, n.d.). Sentinel 2 is a Europe satellite that provides high resolution images. The MSI Level 2A product was done with atmosphere correction and has multiple resolution options. In this study, the spatial resolution of 20 m was selected, which has a total of 10 bands. This study chose to use Band 8a (VNIR), Band 11 (SWIR) and Band 12 (SWIR) because this combination can highlight the burnt area effectively. In the research conducted by Seydi et al (2022), assigning RGB to this band combination will make the burnt areas shown in yellow, which is easy to be differentiated from the blue background (non-burnt areas). These three bands were stacked together using QGIS 3.26.2 (Welcome to the QGIS Project!, n.d.). For the study areas, this study chose two wildfire events in California, they are the August Complex Fire in 2020 (Figure 1) and the Mosquito Fire in 2022 (Figure 2). The former was used as a training sample while the latter was the testing area. The training data was acquired from December 3 to December 10, 2020, and the image size is 7680 × 5760 pixels. The testing data was acquired on October 21, 2022, and the image size is 1536 × 1536 pixels. Both training and testing data have a spatial resolution of 20 m and use the UTM zone 10N projection. This study chose two different events for training and testing to prevent overfitting. Due to some factors like weather condition, time, seasonal change, etc, the intensities of the histograms of the two images do not match. Before feeding into the algorithms, histogram matching was performed on the testing data regarding the training data to increase accuracy.
For the first part, this study chose to use three machine learning algorithms. The training samples were random points generated for both training and testing data using QGIS. They were labeled with four different classes with a unique value from 0 to 3, with 0 being not burnt at all and 3 being extremely burnt. Classes 1 and 2 are slightly burnt and moderately burnt, respectively. The classes were identified based on visual interpretation for the original image, with the darker tone interpreted as more severely burnt. In addition to that, this study performed a pixel-wise classification (semantic segmentation). There are two different types of classification methods: pixel-wise and patch-wise. Pixel-wise classifies each pixel into one class, while patch-wise only has labels on each patch (Guo et al., 2018). This study chose to use pixel-wise classification because it generates classified maps that can be used for visual examination.

The first method was Random Forest (RF). The RF model, proposed by Breiman (2001), is an ensemble learning technique for classification and regression. RF model involves generating
multiple decision trees that work together to classify data. To construct each tree, approximately two-thirds of the input data is used, and the remaining third (known as the out-of-bag or OOB data) is set aside for validation purposes. As the model is being developed, it uses the Gini impurity criterion to estimate the significance of various factors. To compute the Gini impurity, the model randomly alters the values of a given factor (referred to as "m") in the OOB cases, while leaving all other values intact, and tracks the resulting impact on the tree (Jaafari & Keesstra, 2019). The final result is generated by taking the majority vote among every decision tree. The RF model was built using the built-in package made by Scikit-learn (Sklearn.Ensemble.RandomForestClassifier, n.d.), which is a well-known machine learning library for Python. The hyperparameter for RF model is the number of trees, which was set to 50 as the optimized value for this study. This optimized value was found by applying 10-fold cross validation.

The second method selected was eXtreme Gradient Boosting (XGBoost). This model was proposed by Chen and Guestrin (2016), and it is an efficient and scalable implementation of gradient boosting framework that includes efficient linear model solver and tree learning algorithm (Chen & He, 2015). XGBoost works similarly to the RF model, but it is a sequential model, which means it will generate a new decision tree based on the previous result, instead of randomly generating. Thus, it should perform better than RF in theory. Rather than Scikit-learn, this model was built using XGBoost library (XGBoost Documentation — Xgboost 1.7.5 Documentation, n.d.), since Scikit-learn doesn’t have a built-in function of it. This model needs to specify the number of classes, which is 4 in this study. This ensures that the model's outputs are tailored to the specific data set under analysis, allowing for more accurate predictions and
conclusions. There is another hyperparameter named objective which was specified as “multi:softmax”, which is the proper parameter used for multiclass classification.

The third model was Support Vector Machine (SVM). This model was first proposed by Vapnik (1999). Although it’s been in existence for a long time, it still performs well today. SVM uses a statistical learning theory and the structural risk minimization principle to separate two classes with a linear hyperplane (Jaafari & Keesstra, 2019). SVM was also implemented using Scikit-learn (sklearn.svm.SVC, n.d.). Since it is a relatively old method, Scikit-learn has complete support for it. Noted that SVM was originally used for binary classification, the function used was modified by combining multiple binary classification to accomplish multi-class classification task. There are two important hyperparameters for SVM. The first one is the kernel. This is setting the function that is used to separate two classes. “Linear” was selected for this study because it was tested to be the best one. The other one is the regularization parameter, denoted as “C”. This refers to the strength of regularization. It is inversely proportional to C and has to be a positive float. The optimal value was set to 0.0005, which was also found by applying 10-fold cross validation. Noted that SVM is the most time-consuming machine learning method compared to the other two models.

For the second part, this study chose a deep learning model called “U-Net”. U-Net was first proposed by Ronneberger et al. (2015) based on Convolution Neural Network (CNN), and it was initially applied for segmenting medical images. U-Net is famous for its high accuracy for semantic segmentation, and it can be used for classifying wildfire events. The advantage of U-Net is that it only needs a small batch of data to produce accurate segmentation results (Fan et al., 2022). Before building the model, the data have to pass some pre-processing steps. Since U-Net
was invented to segment images, it won’t take random points as training samples. In this case, this study applied a technique called “weak supervised classification” (Fan et al., 2022). This method performed a supervised classification using the best-performed machine learning (SVM in this case) model by training with testing samples and predicting on the same image to generate a mask. This mask has all pixels labeled with a unique class and will be used to train the deep learning model. Both the images and masks were divided into $256 \times 256$ patches before feeding into the model using ArgGIS Pro 3.0.4 (2D, 3D &amp; 4D GIS Mapping Software | ArcGIS Pro, n.d.). To get the optimal result, focal loss was chosen for the loss function, and “Adam” for the optimizer. Focal loss was used because it performs better when dealing with imbalanced data. “Adam”, shorts for adaptive moment estimation, is the most well-known optimizer for deep learning models that can automatically adjust the learning rate and work with loss function to minimize the error.

For the experiment environment, all the models were built on Python 3.10.11 using Google Colab (Google Colaboratory, n.d.) because it offers a free GPU, which is essential for training deep learning models. Some important packages including GDAL 3.3.2, scikit-learn 1.2.2, xgboost 1.7.5, tensorflow 2.12.0, keras 2.12.0 and focal-loss 0.0.7 were also used in the code.

3. Results

The first part of the result is visual comparison. Figure 3 represents the ground truth map, which was generated via SVM. In all the output maps, each pixel has a unique value from 0 to 3 representing the level of burnt, with 0 being not burnt at all and 3 being extremely burnt. For the other four outputs, they all outlined the basic shape of the wildfire event, but they performed differently for each individual class. Figure 4 and Figure 5 are results for RF and XGBoost.
Their background (class 0) looks a little bit noisy compared to the ground truth, which means that they falsely classified some non-burnt pixels as burnt pixels. Looking at Figure 6 and Figure 7, apparently, they did better with class 0, with a clear background and fewer false-classified pixels. By looking at these four outputs, SVM and U-Net performed better than the other two, but it is hard to tell visually which one performed the best.
To address uncertainty in qualitative visual inspection, Table 1 was made to provide quantitative comparison. This table includes some important accuracy metrics, which will be discussed one by one. Those metrics are all calculated using the confusion matrix (Mohajon, 2021). “A confusion matrix is a tabular way of visualizing the performance of your prediction model.” (Mohajon, 2021). There are two pairs of important concepts here: True/False and
Positive/Negative. True or False means if the model correctly makes the prediction or not.

Positive or Negative refers to different classes. Since this is a multiclass classification, we have to focus on one class for better understanding. In this study, if we focus on class 3 (extremely burnt), it will be the positive class. And the other three classes will be negative classes.

Combining them together, we have four values. True Positive (TP): correctly classified positive class; False Positive (FP): falsely classified positive class; True Negative (TN): correctly classified negative classes; False Negative (FN): falsely classified negative classified. Ideally, we want to maximize TP, TN and minimize FP, FN.

Table 1. Quantitative accuracy metrics

<table>
<thead>
<tr>
<th></th>
<th>RF</th>
<th>XGBoost</th>
<th>SVM</th>
<th>U-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>0.85</td>
<td>0.87</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td><strong>Precision</strong></td>
<td>0.48</td>
<td>0.75</td>
<td>0.60</td>
<td>0.64</td>
</tr>
<tr>
<td><strong>Recall</strong></td>
<td>0.42</td>
<td>0.56</td>
<td>0.59</td>
<td>0.81</td>
</tr>
<tr>
<td><strong>F1 score</strong></td>
<td>0.44</td>
<td>0.62</td>
<td>0.59</td>
<td>0.71</td>
</tr>
<tr>
<td><strong>Mean IoU</strong></td>
<td>0.34</td>
<td>0.49</td>
<td>0.51</td>
<td>0.57</td>
</tr>
<tr>
<td><strong>Class 0 IoU</strong></td>
<td>0.90</td>
<td>0.89</td>
<td>0.96</td>
<td>0.93</td>
</tr>
<tr>
<td><strong>Class 1 IoU</strong></td>
<td>0.10</td>
<td>0.31</td>
<td>0.25</td>
<td>0.39</td>
</tr>
<tr>
<td><strong>Class 2 IoU</strong></td>
<td>0.14</td>
<td>0.40</td>
<td>0.41</td>
<td>0.52</td>
</tr>
<tr>
<td><strong>Class 3 IoU</strong></td>
<td>0.22</td>
<td>0.35</td>
<td>0.40</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Mohajon (2021) also discussed these metrics in his article. The first metric, accuracy, is calculated by (TP+TN)/(TP+TN+FP+FN). It is essentially the overall accuracy of the model without considering any individual classes. Since the dataset for this study is imbalanced, most pixels will be classified as background (class 0). Thus, the models can easily achieve high accuracy because the background is the easiest to be classified. The next metric, precision, is calculated by TP/(TP+FP). It indicates the proportion of positive predictions was actually correct.
While recall is calculated by TP/(TP+FN) and it refers to the proportion of actual positives was predicted correctly. Since we usually care about both precision and recall, people merged them together to get F1 score. F1 score is calculated using *Equation 1.* Note that the precision, recall and F1-score were calculated as macro-averaged, which means they were calculated for each class individually and then takes the unweighted mean. This will better reflect the accuracy of every class.

\[
F_1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}
\]

*Equation 1*

Besides all the above, IoU was introduced because it is useful for multiclass semantic segmentation. IoU stands for Intersection over Union. In this case, it represents the overlap between the predicted pixels and the labeled pixels divided by the sum of predictions and labels (Hasty GmbH, n.d.) (see *Equation 2*). It performs similar to the accuracy metric. But what makes it unique is that it can be computed for each individual class, which makes it simple to quantify the accuracy for each class separately. And the mean IoU is calculated by simply taking the mean of the IoU values for every class.

\[
J(A, B) = \frac{|A \cap B|}{|A \cup B|}
\]

*Equation 2*

*Table 1* concludes all the metrics. By looking at it, we can see that the values for most metrics increase from RF to U-Net, which represents the order of these models’ overall
performances. Next, we are going to look into each metric one by one. First of all, U-Net and SVM both got the highest accuracy, which means they both have great overall performances. But accuracy cannot reflect the true situation in this multiclass problem. Next, U-Net had the highest precision and recall, which indicates that it did really well at classifying positive classes. So, U-Net also got the highest F1 score. U-Net has a higher recall than precision, which shows that it did better at classifying actual positive pixels than making a correct positive classification. In other words, U-Net could correctly classify a certain labeled class, but it had a higher chance of classifying other classes as this certain class. In addition, U-Net has the highest mean IoU as well, which indicates that it did the best job at classifying every class. Examining each class, we can find that class 0 was the easiest category to be classified, because most pixels belong to this category. In contrast, class 1 was the hardest category to be classified, which may be due to the small number of training samples as well as the similarity between the three selected bands.

U-Net is expected to get the highest accuracy because it is deep learning. But besides U-Net, SVM got the best performance among the three machine learning models. SVM, the oldest algorithm that was proposed in 1999, generated the accuracies close to the U-Net model. This may indicate that a simple model can also generate great results. In terms of RF and XGBoost, they have more complex mechanisms, but they didn’t perform well in this study.

4. Discussion

Although U-Net had the best performance, it was also the most time-consuming model. It took 19 seconds to train one epoch using the free GPU offered by Colab. This study trained 50 epochs to get the optimal result, which ended up taking 15 minutes. The training time can be greatly decreased by using a better GPU. Same thing for the SVM model, which took a few
minutes to train. This is really slow in the machine learning group, in which RF and XGBoost only need a few seconds to train. Thus, this study shows that good performance requires long training time and better equipment.

This study has some limitations. The most significant one was the imbalanced dataset. Except for the focal loss function that is optimized for imbalanced data, none of the other techniques were used to address this problem. This is probably a potential reason that all the models didn’t perform well in the minority class (class 1). There is a “sample_weight” argument that can be applied when compiling the deep learning model, but it won’t work for this dataset because it has more than three dimensions after divided into patches. Another solution is down sampling the majority class and over sampling the minority classes. This study doesn’t try it due to the time limit, but this method may bring bias to the data and may cause overfitting or underfitting to the training data.

There is another thing that can be improved on for the imbalanced dataset: finding a better loss function suitable for multiclass semantic segmentation. The training process of deep learning is essentially minimizing the error of a loss function. It should get better results if using a function that can calculate loss for each individual class. This study tried several different loss functions, including Categorical cross entropy, Dice coefficient, Jaccard coefficient and Tversky, which are loss functions that have different optimizations for imbalanced dataset. Unfortunately, due to the ways they compute the loss or some coding issues, none of them generate a higher accuracy than focal loss. What’s more, some hyperparameters may not be optimal and need further adjustment, such as the learning rate, batch size and number of epochs.
The next step of this research will be exploring a new variety of U-Net, called “Bi-Temporal U-Net”. This approach uses both pre-wildfire and post-wildfire images to train the model and make predictions. Since comparing between pre and post fire images will highlight the burnt areas, this model should perform better than the traditional U-Net. Another idea is using pre-trained U-Net models. Found some models that are already trained on some dataset, mostly medical images, and then train them on the wildfire dataset. Since they already learned how to classify images, it can have better performance on certain data for this research. And this is an approach for training a small dataset.

5. Conclusion

In conclusion, the implementation of pixel-wise machine learning and deep learning methods on multi-class wildfire mapping has shown promising results in accurately detecting and classifying different levels of burned areas. The results indicate that U-Net did the best job at classifying wildfire events, while SVM had the best performance among machine learning algorithms. U-Net is the most time-consuming model due to the nature of deep learning. The use of high-resolution satellite imagery combined with advanced machine learning and deep learning algorithms has enabled the creation of detailed and accurate wildfire maps, providing valuable information for wildfire management and prevention efforts. However, challenges such as data availability, data quality, and the need for continuous updating of the models to adapt to changing environmental conditions and fire dynamics remain. Therefore, ongoing research is needed to further refine and optimize these methods for more accurate and efficient wildfire mapping. Overall, the use of pixel-wise machine learning and deep learning methods has the potential to
greatly improve wildfire mapping and management, leading to more effective and timely responses to wildfire events.

6. Acknowledgements

The author thanks Dr. Qunying Huang from the Department of Geography, University of Wisconsin-Madison, Dr. Wei Luo and Dr. Alex Haberlie from the Department of Earth, Atmosphere and Environment, Northern Illinois University for their support and instruction.
References


https://doi.org/10.1023/a:1010933404324


Hasty https://doi.org/10.3390/rs14153591


https://doi.org/10.3390/ijgi7030110


Welcome to the QGIS project! (n.d.). https://www.qgis.org/en/site/