Classification Of Damaged Vegetation Areas Using Convolutional Neural Network Over Unlabelled Sentinel-2 Images

Samiha Haque, Nazibur Rahman and Moin Mostakim
Dept of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
Email:{samihahaque96@gmail.com, nazib841@gmail.com, mostakim@bracu.ac.bd}

***Abstract*—Several researches have been made in recent years which use Convolutional Neural Networks (CNN) to classify multispectral images. However, a scarcity of labelled satellite images makes it difficult to classify damaged forest areas using pre-trained CNNs. If any new destruction hits the forests, then there will be a lack of labelled satellite data and manually labeling the satellite images will be inefficient and will ultimately fail the purpose of rapid reforestation. Hence, this research aims to label the unlabelled Sentinel-2 satellite images by using a pre-trained ResNet-50 to obtain flattened feature maps from difference images of disaster struck vegetation areas followed by applying K-Means clustering algorithm on them. Thus, a complete labelled dataset is produced which is used to train two CNN models to classify the areas into undamaged and damaged classes for Model-1 and undamaged, mildly damaged and severely damaged classes for Model-2. Model-1 achieved 89.77% test accuracy and Model-2 achieved 85.69% test accuracy. Model-1 recorded 89.77% in micro f1-score, 76.99% for kappa score and 0.10 for overall error and Model-2 had 85.69%, 72.87% and 0.14 values in micro f1-score, kappa score and overall error respectively.**

***Index Terms*—Convolutional Neural Network, K-Means, Change Detection, Multispectral Image, ResNet-50, Sentinel-2**

I. Introduction

There have been many studies in the field of land classification from images of the Earth's surface using Convolutional Neural Network (CNN). The key properties that make CNN popular in the field of Deep Neural Networks are: it reduces the number of connections, it shares weights on edges, it introduces pooling and it uses several layers [1]. For instance, the authors classified aerial imagery into seven land cover classes using CNN in [2]. Whereas, comparative studies between different CNN models is carried out in [3] (1D CNN and 2D CNN) and [4] (SVM, 1D NN, 1D CNN, RNN, 2D+1D CNN and 3D CNN) which classify land cover and crop types or objects using either Landsat-8 and Sentinel-1A satellite images or hyperspectral image datasets. Reference [5] built an unsupervised deep learning classification framework using Deep Belief Network (DBN) for classification using two types of satellite images (SAT-4 and SAT-6) made of red, green, blue and NIR bands. Their results showed that their classifier performed better compared to a CNN and SDAE. They also stated that unsupervised learning is better for satellite image classification.

Many researches also aim for a solution to classify and locate destroyed vegetation areas. Reference [6] proposed a ternary change detection framework on unlabelled SAR images using sparse autoencoder (SAE) to produce feature maps upon which they used fuzzy c-means (FCM) clustering to obtain pseudo labels to complete their dataset for supervised learning by CNN. Reference [7] uses Caffe framework and AlexNet Network to detect areas which are damaged by flood or landslides by feature extraction. They used Google Earth Aerial Imagery to collect pre-disaster and post disaster images and stacked the RGB pre disaster and post disaster images on top of one another to create the six channel input. Whereas, the authors used CNN to perform binary classification of fire prone forest areas based on the spectral indices: NDVI, NDWI and PSRI of Landsat8 satellite images in [8]. Another research paper detects damaged forest areas by windthrow using U-Net CNN architecture over multispectral aerial remote sensing data [23]. However, they used 2 full 10,000x10,000 pixel photos for testing. It can be noticed that, pretrained CNNs are common to use in these classifications. A comparative study carried out on the accuracy of feature extraction over SAT4, SAT6 and UCMD datasets by pretrained ResNet-50, VGG19, GoogleNet and Alexnet disclosed that ResNet-50 performed the best among the others with highest accuracy scores in [10]. Reference [11] used pretrained ResNet-50 to analyze urban patterns in different ways and found best results when ResNet-50 was tuned to the new dataset before predictions.

Reference [12] is a systemic survey of modern change detection algorithms which is carried out on raw images. The authors of another paper describe different change detection methods that are most widely used over multitemporal remote sensing images [13]. The notable information for us was the algebra based method which includes: image differencing, change vector analysis (CVA) and image rationing which all require thresholding for change detection.

This research aims to build two CNN models to classify vegetation damage levels caused by fires using unlabelled Sentinel-2 satellite images and hence enable rapid reforestation of any fire disaster struck forest area using the already trained CNN models. The objectives of this research are to:

- Produce difference images, DI, from unlabelled bi-temporal multispectral satellite images of recently damaged vegetation areas.
- Prepare a labelled dataset using ResNet-50 over the DI for feature extraction and K-Means algorithm for clustering similar flattened feature maps.
- Construct two distinct CNN models to classify the dataset into damaged and undamaged classes by Model-1 and undamaged, mildly damaged and severely damaged classes by Model-2
- Estimate the amount of each class of damage.

II. Data Collection and Preprocessing

We chose 12 forest areas which are damaged due to large forest fires and two images of the same geometrical location, before and after the destruction, are downloaded from Copernicus Open Access Hub [13]. We chose the maximum resolution i.e., 10m, in order to capture maximum spatial details and also because it contains the Red (R), Green (G), Blue (B) and Near Infrared (NIR) bands. Each image is 10980x10980 pixels representing an area of 100x100 km^2 and does not contain any labels. The images are also pre-processed by scene classification and atmospheric correction [14]. Geographic corrections are not required on the images since both the before and after images have the same spatial coordinates.

A. Band Selection

Bands which distinctly accentuate the abundance or lack of vegetation in a particular region are selected. Red and green bands play a major role in indicating vegetation. Near-Infrared bands reflect chlorophyll well and thus, damaged vegetation will comparatively have low reflectance than healthy vegetation. Blue bands are seldom used for detecting greenery from healthy plants since they do not contribute to the chlorophyll reflectance. Therefore, a three channelled (three dimensional) image is created, consisting of Green (G), Red (R), and Near-Infrared (NIR) bands in the Tiff format. This combination of three bands display a red or pink color to indicate greater chlorophyll reflectance i.e., healthy vegetation [15]

B. Band scaling

Each of the before and after images, from the 12 pairs of images, are scaled using intensity normalization method [11]. For $\mathrm{B}_{\mathrm{ij}}^{\prime\mathrm{k}}$, ij represents the coordinate of a pixel and k is the band number of any band B. The μ^{k} and σ^k represent the mean and standard deviation of k^{th} band.

$$B_{ij}{'}^{k} = (B_{ij} - \mu^k)/\sigma^k \tag{1}$$

This ensures better comparison between the bands of the before and after images since all the bands will have approximately equal scales and units. As stated by [16], a difference image, DI, is created by Image Differencing method which is the pixel wise subtraction of the scaled bands of the after image, Xa_{ij}^k, from the corresponding scaled bands of the before image, Xb_{ij}^k. The scaled G, R and NIR bands are stacked on top of one another to create a three channelled image with G written as band-1, R written as band-2 and NIR written as band-3.

$$DI = {Xb_{ij}}^k - {Xa_{ij}}^k \tag{2}$$

This DI captures the pixel wise change in values which represent either unchanged, positive or negative change in vegetation. Each DI is a three dimensional "change image" with a size of 10980 x 10980 pixels. Using this method, we obtained 12 DI from the 12 pairs of before and after images. Harrietville DI is shown in Fig. 1c as an example. After visualizing and comparing each DI with the corresponding original before and after images, three distinct colors were noticeable : red/pink, teal, gray/ashy green where the red/pink patches of land represent loss of vegetation, the teal patches represent either new vegetation or land surface changes and ashy green patches represent unchanged land.

C. Making Image Crops

The 12 DI obtained are unlabelled and large in size. Each DI is cropped into approximately 8100 smaller three dimensional image crops of 122x122 pixels representing approximately 1.235 km^2. Image crops which contained problematic sections such as, water bodies and random clippings due to the rotation of the satellites, were cropped out. A total of 85,860 image crops of 122x122x3 pixels were obtained.

Labelling these 85,860 image crops manually is inefficient and will lead to a large number of inaccurate labels. To avoid this, an unsupervised clustering algorithm called K-Means is used to cluster the image crops with similar pixel values. In order to improve the clustering by K-Means, an important step of extracting features is included before it. Using transfer learning, a pretrained ResNet-50 is used to extract feature maps of N^{k}x122x122x3 image crops, where N^{k} represents the total N number of crops of the k^{th} DI. The one dimensional flattened feature maps of every DI outputted by ResNet-50 are later inputted into the K-Means to cluster similar flattened feature maps based on similar features. The reason for using ResNet-50 is the ability of the network to learn complex features robustly [1], the flexibility of the code structure and its consistent and better results when compared to other architectures in previous researches [9] and [10]. In this research, we modified the code in order to input our 122x122x3 size images and also to obtain flattened feature maps as outputs from the second last layer of ResNet-50 and therefore, the flattened outputs of each DI have the shape (N^{k}, 32768). TABLE I shows the details of each image.

D. K-Means

Our goal is to build two models: one for classifying only undamaged and damaged land and another for classifying lev-

TABLE I
DETAILS OF IMAGE CROPS

Name	Before date	After date	Image crops
Bolivia	09/07/2019	16/11/2019	8100
Bombala	30/04/2019	15/03/2020	6030
California	25/09/2019	29/10/2020	5850
Canberra	31/03/2019	14/04/2020	8100
Chiang Mai	10/01/2020	19/04/2020	8100
Harrietville	13/01/2019	27/02/2020	8100
Kangaroo Island left	20/10/2019	09/02/2020	5130
kangaroo Island right	23/08/2019	10/03/2020	8100
Salem	21/07/2020	29/09/2020	5940
Tucson	24/05/2020	11/10/2020	8100
Victoria	30/04/2019	11/09/2020	6210
New South Wales	26/03/2019	04/02/2020	8100

els of damage within the damaged areas. Hence two datasets with two sets of labels are required.

a) K-Means for Model-1: The labels for Model-1 dataset must contain two values, one for undamaged areas and another for damaged areas. In order to ensure the correct number of clusters K, we applied the elbow method [17]. The sum of squared errors (SSE or inertia) plotted against the number of clusters displayed two distinct elbows (one at 2 and another at 3) for NSW, Bolivia, Chiang Mai, Victoria, Canberra, Salem, Bombala and Harrietville, only one distinct elbow at 2 for Kangaroo Island left side, Kangaroo Island Right side and Tucson and 2 elbows (one at 2 and another at 4) for California. Since the labels are numeric values between 0 and m, an image I, which is a visual representation of the clusters, is constructed using the labels. This helps to interpret which numeric value corresponds to which flattened image crop, each time after the algorithm is run. An algorithm of the visual representation process is given in Algorithm 1. After visually comparing I with the corresponding before-after images and DI, it was found that K=2 for both left and right sides of Kangaroo Island and K=3 for the rest of the ten areas produced consistent results. It was further observed that the red/pink patches on the DI, which represented "damaged" vegetation, were always consistent with one of the labels. The other two labels which represented unchanged regions and new growth of vegetation or land surface changes were combined under one label called "undamaged" vegetation. In order to bring about consistency, labels corresponding to "undamaged" areas in each dataset are changed to value 0 and labels corresponding to "damaged" areas are changed to value 1 across all flattened outputs.

b) K-Means for Model-2: Model-2 dataset must have three categories: undamaged, mildly damaged and severely damaged areas, so a second K-Means is carried out only on the damaged flattened outputs clustered by the first K-means. Similar to Model-1, the sum of squared errors (SSE or inertia) plotted against the number of clusters showed a single distinct elbow at 2 for Chiang Mai, Kangaroo Island right side (a milder elbow than the rest), Kangaroo Island left side, California, Victoria, Bolivia (a milder elbow than the rest), Harrietville and Canberra, two distinct elbows (one at 2 and another at 3) for Salem and Tucson and extremely mild (close to none) elbow for NSW and Bombala. After running Algorithm 1 and visually comparing I with the corresponding before-after images and DI, it was found that K=2 for all the twelve areas produced consistent results. The "undamaged" labels are obtained using Model-1 label set. Again, by close comparison of I with the corresponding before-after image and DI, it was observed that "severely damaged" labels were produced by a cluster which was consistent with heavily brown burnt ground in after images and "mildly damaged" labels were produced by another cluster. Once more, in order to bring about consistency in the labels, "undamaged" labels in each dataset were kept at value 0, "mildly damaged" labels were changed to value 1 and "severely damaged" labels were changed to value 2 across all flattened outputs.

In the visual representation, I, of each K-Means output for Model-1, the damaged areas are given a red color and the undamaged areas are given a blue color (unchanged land) and green color (new vegetation) and for Model-2, the undamaged areas are given a black color, the mildly damaged areas are given a yellow color and the severely damaged areas are given a red color for ease of visualization and representing the levels of damage. Harrietville is used as an example in Fig. 1d and Fig. 1e

c) Algorithm for visualization: Using the Algorithm 1, crops which are in the same cluster will have the same label number, m, and therefore will be represented using the same color. In doing so, when the visual representation is plotted and displayed in R rows and C columns, m distinct colors form patches in the visual image which is a representation of the DI in terms of the labels predicted by K-Means.

Algorithm 1 Kmeans algorithm

$I \leftarrow R$xCx3 `test image with R rows and C columns with 0 pixels`
$L \leftarrow$ `label values`
$P \leftarrow$ `pixel color`
$m \leftarrow$ `label number`
$c \leftarrow$ `0`
for i `= 0 to` R **do**
  for j `= 0 to` C **do**
    if $L_c = m$ **then**
      $P_c \leftarrow Color_m$
    else if $L_c = m - 1$ **then**
      $P_c \leftarrow Color_m - 1$
      .
      .
    else if $L_c = 0$ **then**
      $P_c \leftarrow Color_0$
    end if
    $c \leftarrow c + 1$
  end for
end for

Fig. 1. (a) Before image, (b) after image, (c) difference image DI, (d) Model-1 K-Means output and (e) Model-2 K-Means output for Harrietville

E. Train Test set

The dataset contains 85,860 flattened feature maps with shape (N^k, 32768). The flattened arrays are reshaped to (N^k, 4,4,2048) where 4x4 are the width and height of each crop and 2048 is the number of channels or depth of each crop. Because the ultimate goal of both the models is to be able to predict any given image once they have been trained, tested and evaluated, three images: NSW, California and Victoria are kept for predictions only and not included in the train and test sets. Therefore, the dataset contained 65700 flattened crops to be split into train and test sets. The dataset is split into 70% training set and 30% testing set using random state. The same flattened image crops are used to train and test both models. The train and test label sets for both the models are coded using one hot encoding method [18].

III. Model Architectures

The aim of this research is to build two CNN models which will act as the continuation from the ResNet-50 model. The hyperparameters used in this model are number of filters, learning rate, number of epochs, batch size, number of convolutional layers, number of maxpool layers, number of neurons in dense layers and number of dense layers. Due to lack of hardware resources such as RAM and Secondary Storage, grid/random search algorithm for fine tuning the parameters could not be performed. Therefore, manual hyper tuning was carried out where a single hyperparameter was tweaked by keeping the rest of the hyperparameters fixed followed by training the model, analysing the results obtained and repeating the process with different hyperparameter values till the model reached satisfactory optimal results. Optimum results for Model-1 were produced by keeping 128 filters in the first layer and doubling the filter number in each layer with an exception of keeping 512 filters in the last two convolutional layers. Number of neurons was also doubled starting with 1024 which is twice of 512 (the filter number in the last convolutional layer). This structure was followed because it resembled the doubling of filter numbers in ResNet-50. For Model-2 a similar approach was taken but it was found that keeping 256 filters in the first two convolutional layers and doubling them in the last two convolutional layers produced optimal results with only one dense layer of 1024 neurons. Similar manual tweaking of batch size and learning rate was carried out for both models.

Model-1 takes inputs in the shape 4x4x2048 i.e the same shape outputted by ResNet-50 before the flatten layer was applied. The first convolutional layer contains 128 filters, the second layer is another convolutional layer with 256 filters, the third layer is a maxpool layer which reduces the feature maps' width and height to 2x2, the fourth layer is another convolutional layer with 512 filters and a fifth layer is another maxpool layer which reduces the feature maps' width and height to 1x1 and the final convolution layer also contains 512 filters. After this, the feature maps are flattened followed by two fully connected layers with 1024 and 2048 neurons. ReLU activation function is used after each convolutional and fully connected layer. According to [22], in comparison to Sigmoid and Tanh, ReLU performs better in terms of generalizing and is fast learning which leads to fast computations. Finally, a softmax classifier is used to output two labels where label=0 corresponds to damaged vegetation and label=1 corresponds to undamaged vegetation. Binary crossentropy is used as the loss function because this model classifies the damaged areas into two classes. Model-1 architecture is displayed in Fig. 2.

Model-2 also takes inputs in the shape 4x4x2048 and has similar architecture to Model-1 with differences in the first convolutional layer with 256 filters and only one fully connected layer with 1024 neurons in the end. A softmax classifier is used to classify the feature maps into three categories where label=0 corresponds to damaged vegetation, label=1 corresponds to mildly damaged vegetation and label=2 corresponds to severely damaged vegetation. Here, categorical crossentropy is used as the loss function because this model classifies the damaged areas into three categories. Model-2 architecture is displayed in Fig. 3.

Throughout all the convolutional layers of both models, a kernel size of 3x3 is used and zero padding is used in order to retain the shape of the feature maps due to the already small width and height. For both the models, all the maxpool layers use a window size of 2x2 and Adam optimizer is used to perform optimization with a learning rate of 0.0001 as it converges faster and has improved learning compared to other different types of self-adaptive learning rate optimizers [19]. Early stopping has been used as a regularization method to prevent the models from overfitting [20]. The training

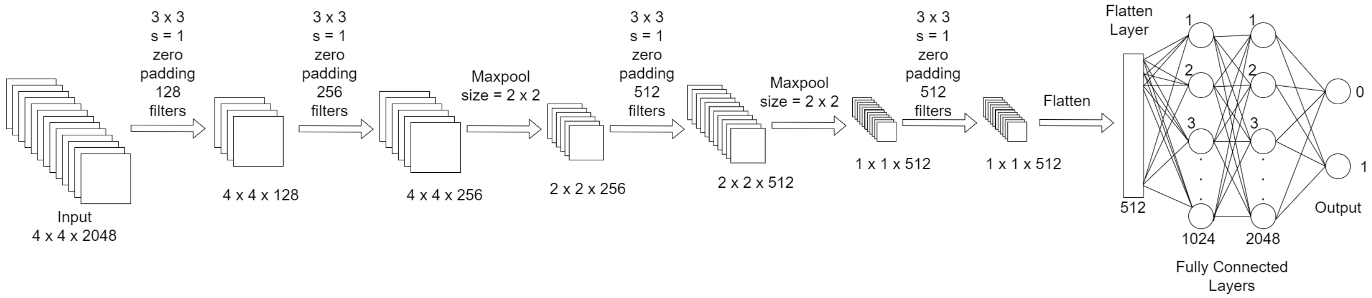

Fig. 2. Model-1 Architecture

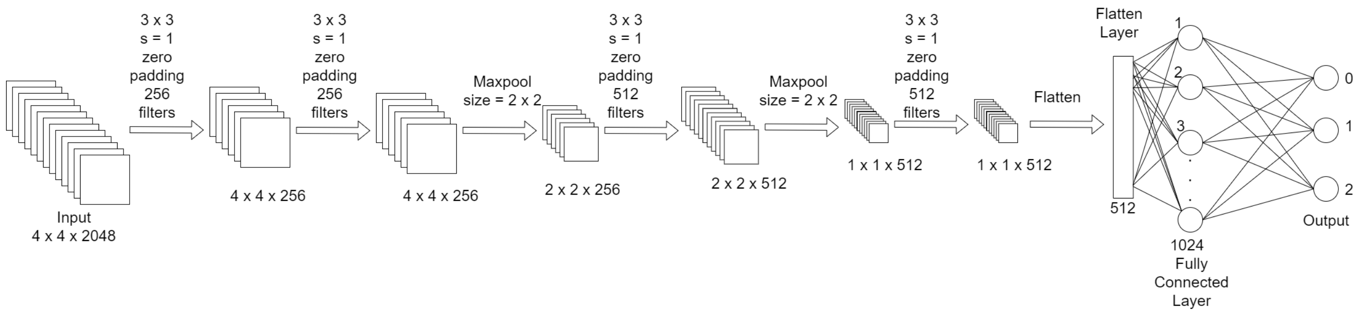

Fig. 3. Model-2 Architecture

data is split further into two sections, one for training the models and the other for validating the models' performance during training. The training data for both the models was split into a 70% training set and a 30% validation set. We used early stopping module of keras with a patience of 4 which means that the training was terminated after 4 epochs of non-improving validation accuracy. After obtaining the approximation of the optimum number of epochs through early stopping, we trained Model-1 over 25 epochs with a batch size of 300 and we trained Model-2 over 24 epochs with a batch size of 300 as well. The best weights were saved and later these weights were used for testing the models and for further predictions.

IV. Evaluation

Both the models were trained and tested on a custom built Google cloud VM instance with 6vCPUs using Intel Skylake CPU platform and 30 GB memory. In this research, both error evaluation and probabilistic accuracy is crucial [21]. In order to evaluate the classifiers qualitatively, we used accuracy, kappa statistic, micro f1-score and overall error (OE) [6].

For evaluating the classifiers as good rankers we used AUC which is the area under the Receiver Operator Characteristic (ROC) curve. This is a metric for binary classifiers and therefore, we used it for Model-1 only.

For evaluating the classifiers as good probabilistic models, we used binary crossentropy during training and testing for Model-1 and categorical cross entropy during training and testing for Model-2.

Model-1 trained for 938s and the best weight was taken at training accuracy of 92.84% with a training loss of 0.1682 and validation accuracy of 90.20% with a validation loss of 0.2450 shown in Fig. 4. Model-1 tested over 19,710 crops each with a shape of 4x4x2048. The precision and recall are equal and this suggests the model is as much precise/accurate as it is robust. Therefore, the quality of reducing errors by Model-1 is satisfactory. Model-1 metrics results are displayed in TABLE IV. The AUC for Model-1 in Fig. 5 is 0.964 which is very close to 1 which suggests that Model-1 is a good separator of damaged and undamaged classes.

Using the number of true positives (damaged areas), the predicted damaged areas are calculated in TABLE II by multiplying the number of crops predicted with 1.235 km^2, i.e., the area represented by each crop.

For the three individual predictions, Model-1 performed quite well on California and Victoria datasets. Victoria has the highest metrics out of the three. However, it performed the worst on NSW. One thing is to be noted that, in between California and NSW, NSW has a greater kappa score than California which indicates that on NSW the model had approximately 8% more agreement with the true labels than California.

Model-2 trained for 1741s, and the best weight was taken at training accuracy of 89.26% with a training loss of 0.2651 and validation accuracy of 85.93% with a validation loss of 0.3602 shown in Fig. 6. Model-2 tested over 19,710 crops each with a shape of 4x4x2048. The same values in both precision and recall of this model suggests that approximately 86% of the time it acts both precisely and robustly. These statistics make Model-2 a quite satisfactory model in terms of reducing errors. Model-2 metrics results are displayed in TABLE IV.

Using The number of predicted labels, the predicted damaged areas are calculated in TABLE III by multiplying the number of crops predicted with 1.235 km^2, i.e., the area represented by each crop.

For the three individual predictions, Model-2 did significantly well in classifying California compared to the other two. However, Model-2 also performed the worst on NSW.

Comparing Model-1 and Model-2 results show that in all metrics, Model-1 outperformed Model-2 by approximately 4%. This is evident since Model-2 is a specialization of Model-1 as it detects levels of damage too and hence more number of classes will correspond to less accuracy comparatively. It can be seen that for the three individual datasets, all scores are in close range of values for Model-1 (except kappa statistic for NSW) but for Model-2, the scores are lagging behind from one another by large margins. Also, both the models performed the worst on NSW dataset. The reason for this might be anomalous labelling of the NSW dataset as K-Means produced loose scattered clusters due to similar values of flattened feature maps. However, Model-2 performed more poorly on the NSW dataset compared to Model-1. On Victoria dataset, Model-1 performed better by a small amount than on California dataset and on California dataset Model-2 performed better by a larger percentage compared to Victoria dataset. The reason for classification accuracy of both the models being lower than most classifiers in this field could be due to the fact that levels of damage contained complex features which were not fully represented by ResNet-50 since it is pretrained on daily object dataset and not satellite images.

TABLE II
PREDICTED DAMAGED AREAS FOR MODEL-1

Data	Damaged Area(km^2)
NSW	4112.55
California	1244.88
Victoria	2145.195

TABLE III
PREDICTED DAMAGED AREAS FOR MODEL-2

Data	Damaged Area(km^2)	
	Mild	*severe*
NSW	1910.545	1259.7
California	443.365	697.775
Victoria	1033.695	306.28

Thus, this could have led to anomalous K-Means clustering which produced multiple wrong labels. Furthermore, some images contained roads, buildings and city parts along with forest areas which could have mixed with other clusters during K-Means.

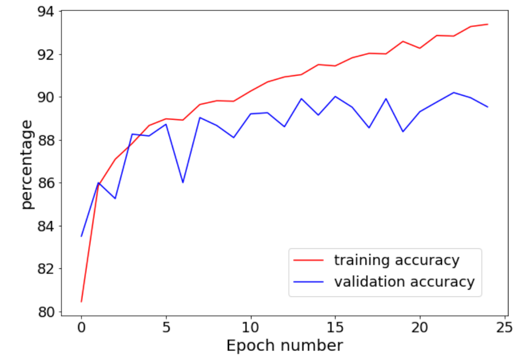

Fig. 4. Model-1 Training vs Validation curve

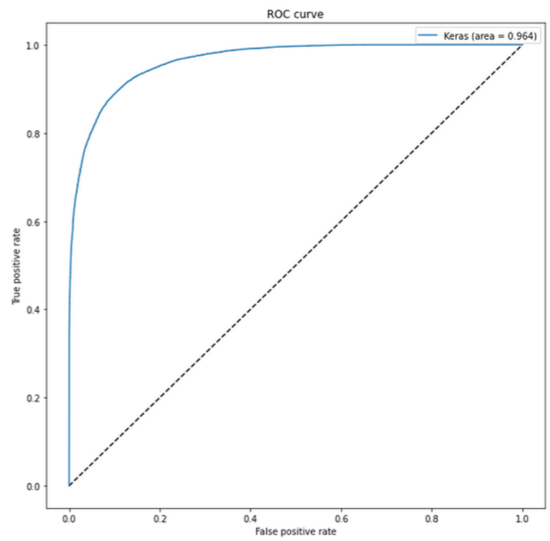

Fig. 5. Model-1 ROC Curve

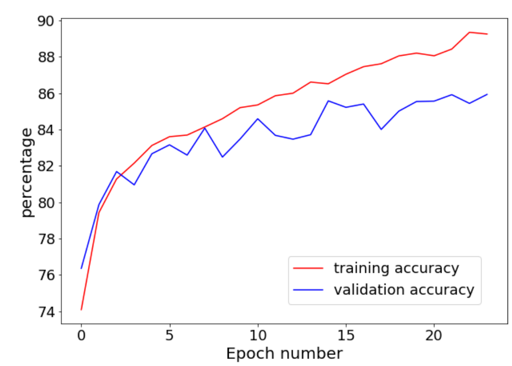

Fig. 6. Model-2 Training vs Validation curve

V. CONCLUSION

In this research, we have constructed two CNN model architectures, Model-1 and Model-2 that classify the flattened feature maps into undamaged and damaged classes and undamaged, mildly damaged and severely damaged classes respectively. We collected before and after Sentinel-2 images of very recent fire-damaged vegetation areas from 12 locations and for each location a three channelled DI was formed using green, red and NIR bands. The difference images are then

TABLE IV
MODEL-1 AND MODEL-2 METRICS RESULTS

Data	Accuracy		Precision		Recall		Micro f1-score		kappa		OE	
	Model-1	*Model-2*	*Model-1*	*Model-2*	*Model-1*	*Model-2*	*Model-1*	*Model-2*	*Model-1*	*Model-2*	*Model-1*	*Model-2*
Test data	89.77	85.69	89.77	85.69	89.77	85.69	89.77	85.69	76.99	72.87	0.10	0.14
NSW	78.15	65.89	78.15	65.89	78.15	65.89	78.15	65.89	56.31	47.17	0.22	0.34
California	79.42	76.75	79.42	76.75	79.42	76.75	79.42	76.75	48.46	48.16	0.21	0.23
Victoria	81.27	70.16	81.27	70.16	81.27	70.16	81.27	70.16	60.10	46.13	0.19	0.30

cropped into 122x122x3 image crops to form a dataset of 85,860 image crops. Among these crops 20,160 of NSW, California and Victoria were separately kept for predicting and the remaining data was split into 70% for training and 30% for testing the models. Our results showed a training accuracy of 92.84% and 89.26% for Model-1 and Model-2 respectively.

For Model-1, the test accuracy reached 89.77% and recorded 89.77%, 76.99% and 0.10 in micro f1-score, kappa statistic and OE respectively. For Model-2 the test accuracy was 85.69% and had 85.69%, 72.87% and 0.14 values in micro f1-score, kappa statistic and OE respectively. Our Model-1 outperformed Model-2 by 4% in all metrics which was expected as the number of classes was higher for model-2.

In the future, other robust pre-trained networks such as VGG19 and AlexNet can be used for a comparative analysis to find the best network for feature extraction from multispectral satellite imagery. Furthermore, Model-1 and Model-2 results can be compared with state of the art Autoencoders or ResUnet results in the future with adequate hardware resources.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[2] M. I. Sameen, B. Pradhan, and O. S. Aziz, “Classification of very high resolution aerial photos using spectral-spatial convolutional neural networks,” Journal of Sensors, vol. 2018, pp. 1–12, Jun. 2018, doi: 10.1155/2018/7195432.

[3] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, “Deep learning classification of land cover and crop types using remote sensing data,” IEEE Geosci. Remote Sensing Lett., vol. 14, no. 5, pp. 778–782, May 2017, doi: 10.1109/LGRS.2017.2681128.

[4] N. Audebert, B. Le Saux, and S. Lefevre, “Deep learning for classification of hyperspectral data: a comparative review,” IEEE Geosci. Remote Sens. Mag., vol. 7, no. 2, pp. 159–173, Jun. 2019, doi: 10.1109/MGRS.2019.2912563.

[5] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, “DeepSat: a learning framework for satellite imagery,” in Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle Washington, Nov. 2015, pp. 1–10. doi: 10.1145/2820783.2820816.

[6] M. Gong, H. Yang, and P. Zhang, “Feature learning and change feature classification based on deep learning for ternary change detection in SAR images,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 129, pp. 212–225, Jul. 2017, doi: 10.1016/j.isprsjprs.2017.05.001.

[7] S. N. K. B. Amit and Y. Aoki, “Disaster detection from aerial imagery with convolutional neural network,” in 2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), Surabaya, Sep. 2017, pp. 239–245. doi: 10.1109/KCIC.2017.8228593.

[8] V. Yaloveha, D. Hlavcheva, and A. Podorozhniak, “Usage of convolutional neural network for multispectral image processing applied to the problem of detecting fire hazardous forest areas,” A.I.S., vol. 3, no. 1, pp. 116–120, May 2019, doi: 10.20998/2522-9052.2019.1.19.

[9] M. A. Kadhim and M. H. Abed, “Convolutional Neural Network for Satellite Image Classification,” in Intelligent Information and Database Systems: Recent Developments, vol. 830, M. Huk, M. Maleszka, and E. Szczerbicki, Eds. Cham: Springer International Publishing, 2020, pp. 165–178.

[10] A. Albert, J. Kaur, and M. Gonzalez, “Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale,” arXiv:1704.02965 [cs], Sep. 2017, Accessed: Jan. 06, 2021. [Online]. Available: http://arxiv.org/abs/1704.02965.

[11] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, “Image change detection algorithms: a systematic survey,” IEEE Trans. on Image Process., vol. 14, no. 3, pp. 294–307, Mar. 2005, doi: 10.1109/TIP.2004.838698.

[12] A. Asokan and J. Anitha, “Change detection techniques for remote sensing applications: a survey,” Earth Sci Inform, vol. 12, no. 2, pp. 143–160, Jun. 2019, doi: 10.1007/s12145-019-00380-5.

[13] Copernicus open access hub. [Online]. Available: https://scihub.copernicus.eu/dhus/.

[14] “User guides - sentinel-2 msi - level-2 processing - sentinel online - sentinel.” https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/processing-levels/level-2 (accessed Jan. 06, 2021).

[15] “Sentinel 2 bands and combinations,” GIS Geography, Nov. 22, 2019. https://gisgeography.com/sentinel-2-bands-combinations/ (accessed Jan. 06, 2021).

[16] S. Mishra, P. Shrivastava, and P. Dhurvey, “Change detection techniques in remote sensing: A review,”International Journal of Wireless and Mobile Communication for Industrial Systems, vol. 4, pp. 1–8, Apr.2017.DOI: 10.21742/ijwmcis.2017.4.1.01.

[17] C. Yuan and H. Yang, “Research on k-value selection method of k-means clustering algorithm,” J, vol. 2, no. 2, pp. 226–235, Jun. 2019, doi: 10.3390/j2020016.

[18] K. Potdar, T. S., and C. D., “A comparative study of categorical variable encoding techniques for neural network classifiers,” IJCA, vol. 175, no. 4, pp. 7–9, Oct. 2017, doi: 10.5120/ijca2017915495.

[19] Y. Wang, X. Wang, and J. Jian, “Remote sensing landslide recognition based on convolutional neural network,” Mathematical Problems in Engineering, Sep. 18, 2019. https://www.hindawi.com/journals/mpe/2019/8389368/ (accessed Jan. 04, 2021).

[20] L. Prechelt, “Early stopping - but when?,” in Neural Networks: Tricks of the Trade, vol. 1524, G. B. Orr and K.-R. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 55–69.

[21] C. Ferri, J. Hernández-Orallo, and R. Modroiu, “An experimental comparison of performance measures for classification,” Pattern Recognition Letters, vol. 30, no. 1, pp. 27–38, Jan. 2009, doi: 10.1016/j.patrec.2008.08.010.

[22] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: comparison of trends in practice and research for deep learning,” arXiv:1811.03378 [cs], Nov. 2018, Accessed: Jan. 04, 2021. [Online]. Available: http://arxiv.org/abs/1811.03378.

[23] Z. M. Hamdi, M. Brandmeier, and C. Straub, “Forest damage assessment using deep learning on high resolution remote sensing data,” Remote Sensing, vol. 11, no. 17, p. 1976, Aug. 2019, doi: 10.3390/rs11171976.