Post-Disaster Building Damage Detection Using Satellite Imagery

G033 (s1988791, s1834500, s1981533)

Abstract

Identifying damaged building after disaster is a vital task for rescue agents in order to supply the first aid to the victims. Conventional methods involve manual extraction of information from aerial images to detect the damaged buildings. However, this method is time-consuming and labor-intensive. Thus, automating the task is inevitable. In this study, we used xBD, a very recent disaster imagery dataset, and developed a multi-path neural network along with a Siamese network for detection of building damage caused due to natural disasters. Both methods performed better than baseline that we established using a very simple CNN architecture. The model that performed better was the dual-path network which had a test set accuracy of 60.10%.

1. Introduction

Disaster management is an emerging topic and has attracted a lot of attention recently due to a high level of increase in the number of catastrophes around the globe. Assessing damage after a disaster should be as fast as possible in order to make emergency response as quick as possible and also, to prioritise the emergency response. The number of rescued victims and fixed damages are strongly related with the urgent detection of the damages in collapsed areas because the detection of damaged buildings also provide useful information for localization of affected populated area.

Many technological pipelines are being used in order to overcome the challenge of the identification of collapsed building in disaster affected area. Remote sensing along with optical satellite imagery, synthetic aperture radar (SAR) and light detection and ranging (LIDAR) technologies are one of the robust and powerful tools to detect the damages and provide useful knowledge to authorities working in these areas. However, gathering useful information from a large scale remote sensing data is not a straightforward task. Conventional methods usually comprise of the manual extraction of information from high dimensional data. Since the response time should be as quick as possible, the extraction of useful information must go beyond manual extraction. Considering the rising number of earthquakes, bushfires and other types of disasters due to global warming and other human-related effects, it is necessary to come up with more efficient and quicker yet robust information extraction methods. Hence, automating the damage detection task by using computational methods is very essential.

Numerous machine learning based methods were developed by using SAR (Shi et al., 2015) (Zhai & Huang, 2016) and LIDAR data (Yu et al., 2017) (Aixia et al., 2016). Support vector machine (SVM) (Li et al., 2010), K-nearest neighbor (K-NN) (Gong et al., 2016), object-based image analysis (OBIA) (Haiyang et al., 2010) are the most common methods in early studies. Most of them provide good accuracy and a reasonable processing time. However, collection of data in the collapsed area is dangerous and hard because these areas are usually hard to access. Hence, a recent breakthrough in deep learning after advancement in convolutional neural networks directed most researchers to apply deep learning methods for disaster management (Ji et al., 2018). Also, the advancement of aerial imagery along with increased usage of unmanned aerial vehicles (UAV) provided access to large scale image databases (Kerle et al., 2020).

Most of researchers who applied machine learning on disaster imagery dataset focus only single disaster images. (Cooner et al., 2016) did a comparative study for building damage detection with satellite imagery of 2010 Haiti earthquake. Their work showed that the lowest error rate was achieved by feed-forward neural network. Another work done by (Ji et al., 2018) succeed 78.6 % overal accuracy on the same disaster, 2010 Haiti earthquake. In 2019, (Xu et al., 2019) used a disaster dataset consisted of 3 different earthquake around world and performed the generalizability of CNN in detecting building damage caused by disasters. Their results showed that the generalization would be better if the training set consist of very small set of examples from that region. Since damaged building detection in disaster imagery becomes urgent topic in recent years, a new dataset (Gupta et al., 2019) which consists of the images from many different disaster types, was proposed very recently.

In this work, we used a balanced version of xBD (Gupta et al., 2019) dataset in order to detect the damage in buildings after a disaster. In this study,we first established a baseline and then developed a multi path neural network and a Siamese network to undertake the task of damage detection. Our main research question was to investigate the performance of different models on the new xBD dataset. On xBD, no substantial research has been published yet, apart from the original work (Gupta et al., 2019). The results of our study showed that our models provided a good accuracy in detecting damage in buildings in the xBD dataset when compared to our baseline.

Figure 1. Pre-disaster imagery (top) and post-disaster imagery (bottom). From left to right: Hurricane Harvey; Joplin tornado; Lower Puna volcanic eruption; Sunda Strait tsunami (Gupta et al., 2019)

The rest of the paper was organized as follows. In section 2, the dataset and task were described. Section 3 focuses on the techniques and methodologies used. In section 4, we talked about the performed experiments and their result. Discussion forms the 5^{th} section of our study. Then, the paper was concluded with related work and conclusion.

2. Data set and task

The dataset we used in our study is a relatively new and large scale dataset called the xBD dataset (Gupta et al., 2019). The data was collected primarily in order to detect changes as well as building damage for human assistance and disaster recovery research. xBD consists of pre and post event satellite imagery which comprises of a variety of disaster events. Till now, xBD is the largest building damage assessment dataset comprising of 850,736 building annotations across 45,362 square km of imagery. Thus, it is pretty evident that it covers a very wide range of disasters (19 disaster types) and geographical locations (15 countries). The importance of this dataset can be interpreted from the fact that understanding the damage caused on buildings from satellite imagery is a very complex task because different disasters would impact and damage buildings in different ways and to different extents. The dataset consists of the following: building polygons, which are granular representations of a building footprint, ordinal regression labels for damage, which rate how damaged a building is on an increasing integer scale, multi-class labels, which relate all the environmental factors that caused the damage seen in the imagery and environmental factor bounding boxes and labels, which are a rough approximation of the area covered by smoke, water, fire, and other environmental factors. All the images for the xBD dataset is sourced from DigitalGlobe (https://www.digitalglobe.com) which provides high resolution imagery at approximately 0.5m ground sample distance. This is more than sufficient labelling. All the annotated buildings have a label describing the extent of damage, it belongs to one of these four categories: No Damage (0), Minor Damage (1), Major Damage (2) and Destroyed (3). Figure 2 gives an elaborate description of the extent of the damage.

Disaster Level	Structure Description
0 (No Damage)	Undisturbed. No sign of water, structural or shingle damage, or burn marks.
1 (Minor Damage)	Building partially burnt, water surrounding structure, volcanic flow nearby, roof elements missing, or visible cracks.
2 (Major Damage)	Partial wall or roof collapse, encroaching volcanic flow, or surrounded by water/mud.
3 (Destroyed)	Scorched, completely collapsed, partially/ completely covered with water/mud, or otherwise no longer present.

Figure 2. An elaborate description of the Damage Levels caused by the Natural Disasters (Gupta et al., 2019)

There are a total of 22,068 images in the dataset and each has a resolution of 1024 x 1024. Train set has a total of 18,336 images and 632,228 polygons whereas test set has 1866 images and 109,724 polygons. Validation set accounts for 1866 images and 108,784 polygons.

The task we are going to explore here is mainly detecting damages on buildings and structures from the satellite images in order to aid workers and mount effective response from the emergency services. Extracting useful information manually from the satellite images is a very sluggish and

labour intensive task because of which machine learning algorithms are used nowadays to automate the task of detecting building damage from the satellite imagery. The original paper (Gupta et al., 2019) used a multi-path model which comprised of a ResNet50 and a shallow CNN model. This model was their baseline (see figure 4). Weighted F1 score was used as the primary metric for evaluating the performance (as the dataset was heavily imbalanced) of the classification model. Owing to the imbalanced nature of the dataset, the model performed really poorly. As a result of this, we pre-process and perform undersampling on the original dataset so as to make the number of classes (ranging from 0 to 4) equal in number. Since, in our study, we use the balanced version of xBD dataset, classification accuracy was used as the evaluation metric. Classification accuracy is defined as the ratio of number of samples classified correctly to the number of total samples.

Another important factor which was observed in study performed by (Gupta et al., 2019) was that class 0 (No Damage) and class 3 (Destroyed) were easily distinguishable whereas it was difficult for their baseline to understand the difference between class 1 (Minor Damage) and class 2 (Major Damage). The two models which were implemented in our work namely Siamese Model and Dual-Path Model on the balanced dataset produced a classification accuracy of 60.80% and 61.74% on the training set respectively and a classification accuracy of 58.97% and 60.10% on the testing set respectively.

The distribution of the new dataset after undersampling is as follows: 80% of the data used as training data (20050 polygons), 10% as validation (2504 polygons), 10% as test (2506 polygons).

The distribution of the classes in the training data : 5033 instances belonging to class (0),4952 instances belonging to class (1),5032 instances belonging to class (2),5033 instances belonging to class (3). A similar distribution as training set is followed in validation and test sets.

Figure 1 gives an illustration of the images before and after a disaster.

3. Methodology

This section covers the details of the technologies and methods used for conducting the experiments necessary for achieving the building damage detection task.

3.1. Convolution Neural Networks (CNNs) and Max-Pooling

CNNs were first introduced in the work done by (LeCun et al., 1998). CNNs are a type of Neural Network that are usually used in tasks that involve images as the input, such as ours. The most important component of a CNN are the convolution layers. An input given to a convolution layer consists of a number of feature maps and ,similarly, the computed output of a convolution layer comprises of a number of feature maps. The number of feature maps in the output is equal to the number of filters in the convolution layer and this number can be different or same as the number of input feature maps. For each convolution layer in a CNN, a convolution operation is performed on the input that each layer receives. Convolution operation could be expressed as equation 1.

$$S(j,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \quad (1)$$

In equation 1, I stands for a two dimensional image and K stands for a two dimensional filter in a convolution layer.m and n represent the width and the height of the filter K. A thing that can be observed from the equation 1 is that after the convolution operation, the size of the resultant image is smaller than the original input. Though, the size of the output could be forced to be same as the input by making use of a suitable padding.

Translational invariance is the principle on which CNNs work. Translational invariance stands for the fact that even though the appearance of an object (in our case, in an image) is modified in some way or the other, it can still be recognized. Translation, size and rotation invariances are some of the most common invariances. It is hard to achieve translational invariance just by the use of convolution layers as the output feature maps in convolution layers tend to be sensitive towards the location of the features with respect to the input image. This problem is dealt with by employing local translational invariance which involves down sampling the output feature maps of a convolution layer such that it is no longer sensitive to the position of a feature in the image. Local translational invariance is achieved by using pooling layers and in our case, max-pooling layer. This is how CNNs are capable of capturing the low level spatial and temporal features in the input data that is fed to them.

3.2. Transfer Learning

Transfer learning (Torrey & Shavlik, 2010) is a technique in deep learning in which a new model's layers are initialized with the weights of a model that was pretrained on a different dataset. It is not necessary for the new model to be exactly the same as the pretrained model but it is necessary that the new model contains at least a part of the pretrained-model. There are generally two ways of training a new model using transfer learning. Under the first approach, the part of the new model that shares the same structure as the pretrained model is initialized with the pretrained weights. When this new model is trained, only those layers that have not been initialized with the pretrained weights are trained (i.e. weights of pretrained part does not change during the training). Under the second approach, just like the first approach, the new model is initialized with pretrained weights. But this time, weights of all the layers change during the training process. For the baseline mentioned in damage detection task of (Gupta et al., 2019), first approach is followed.

3.3. Multi Path Network

A multi path network (Wang, 2015) ,like the name suggests, is a neural network that uses multiple neural network paths for the task of learning the intended information from a given dataset. A multi path network is essentially a combination of multiple neural networks combined together in some manner to achieve a collective task. It is not necessary for every sub-path to have the same structure and also, it is not necessary for every sub-path to merge at the same point in the model. There are numerous ways of merging two sub-networks. Some of the most common ways are: element wise addition, subtraction or multiplication of the merging layers and the other possible way is to simply stack the two merging layers to produce a new layer whose length is equal to the combined length of the merging layers. In the element-wise operation merger, it is necessary that the merging layers have the same number of elements or nodes.

3.4. Siamese Neural Network

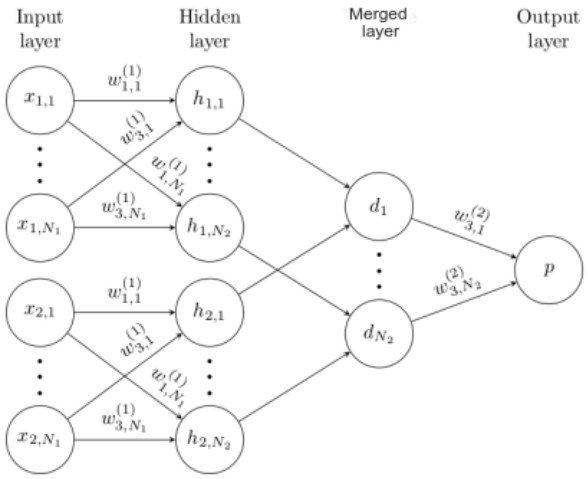

Figure 3. A simple 2 hidden layer siamese network for binary classification with logistic prediction p. The structure of the network is replicated across the top and bottom sections to form twin networks, with shared weight matrices at each layer. This image has been taken from (Koch et al., 2015)

Siamese networks were first introduced Bromley and LeCun (Bromley et al., 1994) in 1990 for the task of signature verfication. It was seen as an image matching task. A Siamese network is basically a multi path network comprising of two sub-paths with some constraints on its structure. The two sub-paths must be exactly the same in terms of structure and also, must share the same weights. One of the most promising use cases was given by (Koch et al., 2015) in which they made use of a Siamese CNN to perform one-shot learning on an image classification task. Figure 3 is a good representative of a Siamese network.

3.5. Optimization

Optimization is the technique which is used to optimize the learning rate of the model during training. In our case the optimizer used was Adam which was first mentioned in (Kingma & Ba, 2014) and showed some promising results. Adam is one of the most popular optimizers in deep learning these days. Even in the (Gupta et al., 2019) , baseline model used Adam optimizer. Adam can be seen as RMSprop (Ruder, 2016) and momentum working in conjunction. Adam makes use of the moving average of first moment and the moving average of second moment in computing learning rates. Adam has two parameters $\beta 1$ and $\beta 2$ which control the rate of decay of the two moving averages.

3.6. Regularization:

In large neural networks, overfitting (Ying, 2019) is almost always an issue. When a neural network overfits, it's performance on training data is really good but that is not the case for unseen data or the test data. A neural network's performance on unseen data is known as generalization performance, and a good generalization performance is always desirable. In our case, generalization performance happens to be the classification accuracy on test set. To counter overfitting, regularization is used. One of the such regularization techniques is dropout. First mention of dropout was in (Srivastava et al., 2014) . In dropout, some of the neurons in the neural networks are randomly ignored with some probability during the training. This probability is known as dropout probability or λ. This reduces the complexity of the neural network and thus, reduces overfitting.

4. Experiments

In order to achieve the task of damage detection on the balanced xBD dataset, the first task was to see if the original transfer learning based baseline model from (Gupta et al., 2019) is sufficient enough or not. This model shall now be referred as *"reference model"* in the rest of our work to avoid any confusion with the baseline set up in later stages of our experiments.The reference model comprised of a multi-path network with two sub-paths (See figure 4).An important aspect about all the models used in our study is that all of these models used categorical cross entropy as the loss function.

The first sub path (*Sub path-1*) in the reference model was a shallow CNN model with three convolution - max pooling layer pairs. Every convolution layer in the network had ReLU (Nair & Hinton, 2010) activation function. The first convolution layer in this CNN had 32 filters with a kernel size of 5X5 and the remaining layers had 64 filters each, with a kernel size of 3X3. The second path (*Sub path-2*)was a ResNet50(He et al., 2016) model pre-trained on Imagenet (Russakovsky et al., 2015) (and hence , transfer learning is used for testing the reference model). Outputs from both the sub paths are flattened before they are merged. In order to merge the layers, the flattened output from both the paths were simply stacked together. The merged layer was followed by three dense layers.First,second and third dense layers had 2024, 524 and 124 neurons, respectively. These dense layers, just like the shallow CNN's layers, had ReLU as their activation function. The last layer of the reference

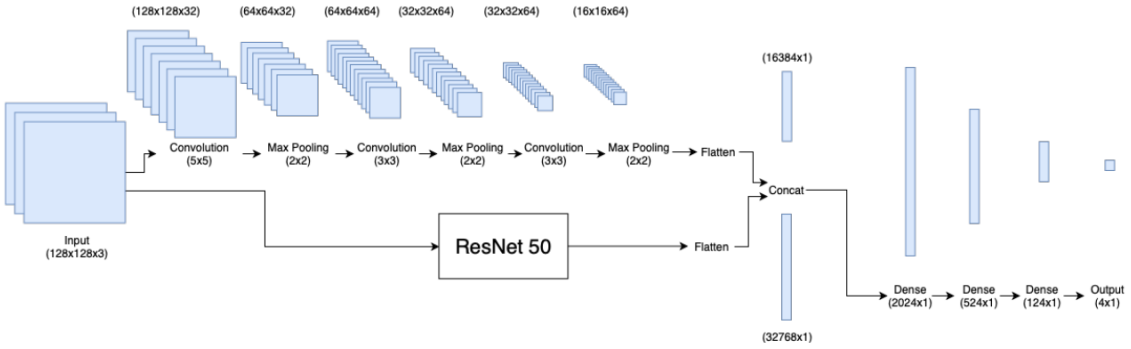

Figure 4. Architecture of the reference model. The input is fed into a pre-trained ResNet 50 as well as a shallow CNN. The outputs of each stream are concatenated and passed into dense layers for classification.The image was taken from (Gupta et al., 2019)

model was a softmax layer.

For both the sub networks, same post-disaster image from the balanced dataset was fed as an input. A key thing to note about the training process of this model is that during the training time the weights of the ResNet-50 model or the Sub path-2 do not change. The original training conditions of Adam optimizer ($\beta 1 = 0.9$ & $\beta 2 = 0.99$), batch size (64),input image size (128 X 128) and learning rate (0.001) from (Gupta et al., 2019) were used. The model was trained for 50 epochs, instead of the original 100 epochs. It was seen that model was severely overfitting. The reason behind training the model for only 50 epochs was to test it's feasibility for the balanced xBD dataset. From figure 5, it can be seen that the validation set accuracy of this model is close to 25% which basically meant that for each input, model was assigning a class at random. This was the case when classification was done for all the four classes but when the two easily distinguishable classes, 0 and 3, were only used for training, even then the model overfit and gave a validation accuracy of 50% which again was equivalent to a random assignment of classes to a given input. We were sure that the reference model was overfitting because of very high training accuracy and a poor validation accuracy (see figure 5).

NOTE: Through out our work the conditions/parameters of Adam optimizer, batch size, input image size and learning rate remain same as the reference model

Based on the above experiment, it was hypothesized that the reference model was too complex for the balanced xBD dataset and hence, a less complex CNN architecture should be able to perform better. In order to validate this hypothesis, the ResNet50 sub-network (or the Sub path-2) of the reference model was removed keeping remaining architecture of the reference model intact.This new model was referred as *Shallow CNN model*. The achieved results did reconcile with our hypothesis. The validation accuracy of the shallow CNN model at the end of the training was 53.85% when classifying all the four classes and 85.70% when classifying the two easily distinguishable classes: 0 and 3. From figure 6, it can be seen that the shallow model didn't really overfit. This shallow model was our baseline (see Tables 1) .

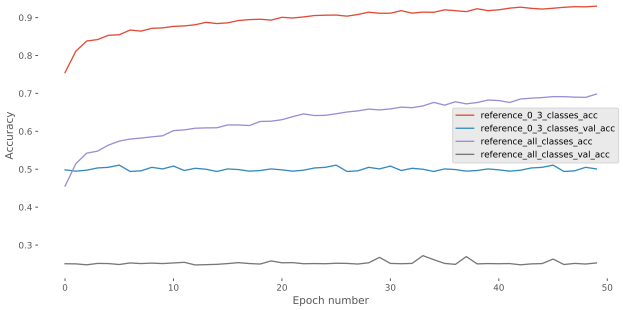

Figure 5. Comparison of the training and validation accuracy for the reference model. "Reference_0_3_classes" represents the model that was trained on only the two easily distinguishable classes. "Reference_all_classes" represents the model that was trained on all the classes of the dataset.

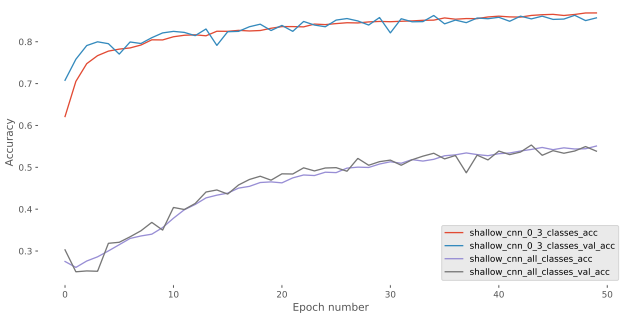

Figure 6. Comparison of training and validation accuracy of the shallow CNN or the baseline model trained for 50 epochs."Shallow_cnn_0_3_classes" represents the baseline model that was trained on only the two easily distinguishable classes. "Shallow_cnn_all_classes" represents the baseline model that was trained on all the classes of the dataset.

Classes	Train Acc.	Valid Acc.
0&3	86.88	85.70
All	55.09	53.85

Table 1. A summary of the training and validation set accuracy for the baseline when it was trained for 50 epochs."Classes" refers the classes of the balanced dataset on which the model was trained.

From our baseline model(s), it was established that a shallow CNN works better in this scenario (validating our last hypothesis) and the complex reference model doesn't. A most plausible explanation to this can be the fact that the input images are just 128X128 in size and the features that have to be extracted by the CNNs are relatively very simple compared to other CNN based image tasks.

Following our observations from the previous experiments, we investigated the effect of replacing the complex ResNet50 (Sub path-2) from the reference model with a shallower sub path. In this experiment, instead of taking same post-disaster image as an input to both the sub paths, a pre-post disaster image pair is taken as input. The motivation behind this new scheme are the facts that the proposed model will have less complexity (therefore, shouldn't overfit as bad as the reference model) and the presence of pre-disaster image in input will act as a reference for the model to compare with post-disaster image (thus, enabling the model to figure out to what extent is the post-disaster image different from the pre-disaster image).

In order to test this new scheme, the reference model was taken and ResNet50 (Sub path -2) was replaced by an exact copy of the shallow CNN sub path (Sub path-1) that was already present in the network. So essentially, this new model is the reference model but with two Sub path -1 and no Sub path -2. The other thing that was different than the reference model was the way the two sub-networks were merged. Element wise addition, element wise subtraction and layer stacking were tested as candidates for merging techniques. It can be seen from table 2 that element wise subtraction was the one that gave better results and also, it has a better validation set performance compared to our baseline model. Thus, it is the new best performing shallow multi-path model and for clarity,this model will be referred as dual-path model in the rest of this work. Though, it can be seen from figure 7 that the subtraction based merging model shows some tendency to overfit.

Model	Train Acc.	Validation Acc.
Subtraction-Multipath	66.11	58.57
Addition-Multipath	24.57	25.12
Stacking- Multipath	53.18	53.28

Table 2. A summary of training and validation set performance of dual-path model when using subtraction,addition and stacking based merging techniques.

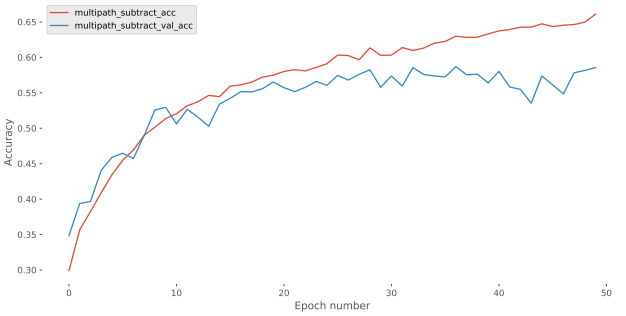

Figure 7. Accuracy Vs Number of epochs trend for the dual-path model based on subtraction merging technique.

In order to deal with the overfitting of subtraction based dual-path model, dropout regularization was selected as the choice of regularization. Since, dropout regularization was to be used, it only made sense to train the dual -path model in excess of the original number of epochs for which the baseline model was trained. Therefore, the number of epochs was increased from 50 to 100. As the number of epochs for which the dual-path model was trained increased to 100, the baseline model was trained again for 100 epochs to make the comparison justified (see table 3).

Model	Train Acc.	Valid Acc.	Test Acc.
Baseline (100 epochs)	60.66	59.02	57.09

Table 3. A summary of the training, validation and test set accuracy for the baseline when it was trained for 100 epochs. The baseline was trained on all the four classes in the balanced dataset.

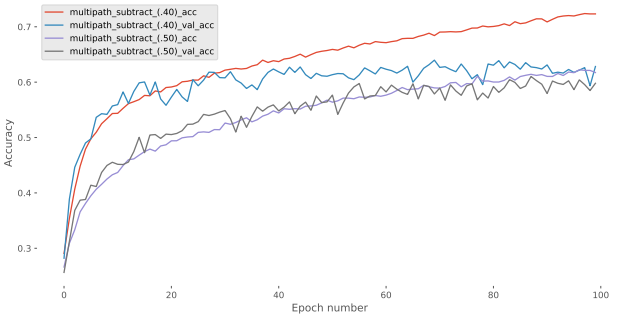

Figure 8. Comparison of training and validation accuracy of subtraction merging based dual-path model."Multipath_subtract_(.40)" represents the model with λ=0.40."Multipath_subtract_(.50)" represents the model with λ=0.50.

The dropout layers were only put after the merged layer and the first dense layer as these layer were relatively way more complex than any other dense layer due to large number of neurons present in them(16384 and 2024 nodes, respectively). In order to fine tune the network, different dropout probabilities ($\lambda = [0.0, 0.10, 0.20, 0.30, 0.40, 0.50]$) were tested. It was observed that the one that performed the best in terms of generalization performance or the test set accuracy was achieved by the model with a λ=0.40 (see table

4). But it can also be seen that the model with λ=0.40 has more overfitting tendency than the one with the λ=0.50 (see figure 8). Therefore, because of this reason, model with λ=0.50 was chosen as the new best performing dual-path model. The results from this experiment can be summarized by the table 4 .

λ	Train Acc. (%)	Valid Acc.(%)	Test Acc.(%)
0.00	75.33	56.72	55.73
0.10	82.84	62.34	59.78
0.20	74.41	58.65	59.62
0.30	71.14	58.24	58.29
0.40	72.33	62.83	60.22
0.50	61.74	59.80	60.10

Table 4. A summary of the train , test and validation set accuracy for the subtraction merging based dual-path model with dropout probabilities given by λ

Now, in addition to the experiments related to dual-path model, we also ran experiments to test if a Siamese neural network with the same architecture as the dual-path network (but with shared weights between the two sub-paths) can perform better than the dual-path network. It was observed that the Siamese network achieved accuracy of 60.80%,58.98% and 58.97% on train,valid and test set, respectively. The performance of Siamese network is very close to the dual-path network. It is also worth mentioning that due to weight sharing, the number of trainable parameters in the Siamese network is smaller compared to the dual–path network(see table 5).

All the experiments that have been discussed until now were ran on 3 Nvidia GeForce GTX 1060 6GB graphics units running in parallel.

Model	Train acc.(%)	Test acc.(%)	Trainable Parameters
Baseline	60.66	57.09	34,405,652
Siamese Model	60.80	58.97	34,347,796
Dual-Path Model	61.74	60.10	34,347,796

Table 5. A summary of all the models that were used for undertaking the damage detection task in the balanced xBD dataset.Along with the training and test accuracy , number of trainable parameters have also been mentioned.

5. Discussion

From the initial experiments conducted using the reference model, which was used in (Gupta et al., 2019) as the baseline, it could be hypothesized that this reference model was too complex for the new balanced xBD dataset. This hypothesis is verified by using only the shallow CNN model (Sub path-1) from the reference model for training on the dataset.For our baseline, the shallow CNN model was run for 50 epochs and the reported train and validation accuracy were 55.09% and 53.85%, respectively. After this experiment, in order to achieve a greater generalization performance or test set accuracy, a multi-path model comprising of two identical networks was used. This model was called "dual-path model".Our initial hypothesis about this model was that it should at least work better than reference model as it was far less complex than it. Our hypothesis was verified by the results achieved (see Subtraction-Multipath in table 2). But it was noticed that this model had a very high tendency to overfit which was rectified by making using of dropout regularization. Since, dropout regularization was to be used, the dual-path model was trained for 100 epochs. This solved the issue of overfitting which is inline with the study conducted by (Srivastava et al., 2014). The final reported test accuracy of this model was 60.10%. Since, the dual-path model was trained for 100 epochs therefore, baseline was re-trained for 100 epochs to keep the comparison just. Then we moved ahead and tried to inspect the effect of weight sharing if it were to be used in the dual-path model. For this part, every condition was kept same as the dual-path model apart from weight sharing (the two sub-paths shared the same weight as seen in figure 3). This model was referred as "Siamese Model" .The test accuracy of the Siamese model was 58.97%.

6. Related work

In spite of emergence of damaged building detection after disasters, there is not enough work in literature. Most of works are based on one type of disaster. Artificial Neural Networks (ANNs) were used succesfully in early studies (Cooner et al., 2016). The results of these studies were promising. Radial basis function neural networks (Li et al., 2016), SVM (Li et al., 2009) and Random forests (Pal, 2005) were quite popular among those studies. However, the performance of these studies remain controversial when it comes to large scale databases (Cooner et al., 2016).

Since the dataset used in this study is quite new, the related works are also very recent. (Xu et al., 2019) proposed a method to build a CNN which could identify collapsed buildings in satellite images. Although it seems that they used different dataset than ours, their dataset was in fact gathered from xBD. In their work, they also performed a novel data generation method. The main contribution of their work is to develop a generalized model for new regions. They also found that the generalization for better accuracy is possible if the model is fine tuned on a small subset of examples from the specified region.

Another interesting work is the one proposed by (Ji et al., 2018). They combined a CNN along with regression based model in order to assess the degree of damage. While CNNs extracted useful features from images, the regression model classified these features into four categories. Due to lack of robust database which include labeled images of damaged buildings, most studies produced their dataset.

In this work, we were inspired by a series of studies in deep learning literature(Wang, 2015) (Bromley et al., 1994). These types of network which comprised of multipath net-

works, utilized for the task such as image matching. To the best of our knowledge, there is not much work done for damaged building detection for this new dataset,xBD. The aim of this study is to apply the multipath networks with different configurations to the balanced xBD dataset in order to detect the damage in buildings with high accuracy.

7. Conclusions

After conducting all the experiments on the balanced xBD dataset using a number of neural network architectures, it could be concluded that for any CNN based architecture , adding more and more layers may not give a satisfactory solution to the problem. For some tasks, a simple-shallow CNN architecture can work better. ResNet50 architecture was making the reference model way too complex but the shallower CNN models worked a lot better. It's not just how a CNN architecture is defined but also, the kind of input the architecture is being fed counts a lot towards generalization performance. When a CNN architecture has access to a reference input image, it can show improvements in it's generalization performance. Techniques like dropout regularization might be very simple but can be effective in improving a neural network's generalization performance. When these techniques were combined together in the two candidate models: dual-path and Siamese model, the test accuracy produced was 60.10% and 58.97%,respectively. This meant that the dual-path model is the best performing model .

One of the other possible ways of increasing the generalization performance of a CNN is increasing the number of filters in the convolution layers,e.g. from 32 to 64 or from 64 to 128. In future, this could be experimented with. Also, hyper-parameters like $\beta 1$ and $\beta 2$ of Adam optimizer, along with the learning rate could be experimented with to see if it's possible to achieve better test set accuracy.One of the interesting aspects that could be additionally tested would be what would have happened if the ResNet-50 (Sub path-2) in the reference model would also be allowed to train it's weights like the other layers.In addition to these, an ensemble technique of classifiers could also be explored.

References

Aixia, Dou, Zongjin, Ma, Shusong, Huang, and Xiaoqing, Wang. Building damage extraction from post-earthquake airborne lidar data. *Acta Geologica Sinica-English Edition*, 90(4):1481–1489, 2016.

Bromley, Jane, Guyon, Isabelle, LeCun, Yann, Säckinger, Eduard, and Shah, Roopak. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pp. 737–744, 1994.

Cooner, Austin J, Shao, Yang, and Campbell, James B. Detection of urban damage using remote sensing and machine learning algorithms: Revisiting the 2010 haiti earthquake. *Remote Sensing*, 8(10):868, 2016.

Gong, Lixia, Wang, Chao, Wu, Fan, Zhang, Jingfa, Zhang, Hong, and Li, Qiang. Earthquake-induced building damage detection with post-event sub-meter vhr terrasar-x staring spotlight imagery. *Remote Sensing*, 8(11):887, 2016.

Gupta, Ritwik, Hosfelt, Richard, Sajeev, Sandra, Patel, Nirav, Goodman, Bryce, Doshi, Jigar, Heim, Eric, Choset, Howie, and Gaston, Matthew. xbd: A dataset for assessing building damage from satellite imagery. *arXiv preprint arXiv:1911.09296*, 2019.

Haiyang, Yu, Gang, Cheng, and Xiaosan, Ge. Earthquake-collapsed building extraction from lidar and aerophotograph based on obia. In *The 2nd International Conference on Information Science and Engineering*, pp. 2034–2037. IEEE, 2010.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ji, Min, Liu, Lanfa, and Buchroithner, Manfred. Identifying collapsed buildings using post-earthquake satellite imagery and convolutional neural networks: A case study of the 2010 haiti earthquake. *Remote Sensing*, 10(11): 1689, 2018.

Kerle, Norman, Nex, Francesco, Gerke, Markus, Duarte, Diogo, and Vetrivel, Anand. Uav-based structural damage mapping: A review. *ISPRS International Journal of Geo-Information*, 9(1):14, 2020.

Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koch, Gregory, Zemel, Richard, and Salakhutdinov, Ruslan. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, Jiaojiao, Du, Qian, and Li, Yunsong. An efficient radial basis function neural network for hyperspectral remote sensing image classification. *Soft Computing*, 20(12): 4753–4759, 2016.

Li, Liwei, Li, Zuchuan, Zhang, Rui, Ma, Jianwen, and Lei, Liping. Collapsed buildings extraction using morphological profiles and texture statistics-a case study in the 5.12 wenchuan earthquake. In *2010 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2000–2002. IEEE, 2010.

Li, Peijun, Xu, Haiqing, Liu, Shuang, and Guo, Jiancong. Urban building damage detection from very high resolution imagery using one-class svm and spatial relations. In *2009 IEEE International Geoscience and Remote Sensing Symposium*, volume 5, pp. V–112. IEEE, 2009.

Nair, Vinod and Hinton, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Pal, Mahesh. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

Ruder, Sebastian. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., and Fei-Fei, Li. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Shi, Lei, Sun, Weidong, Yang, Jie, Li, Pingxiang, and Lu, Lijun. Building collapse assessment by the use of postearthquake chinese vhr airborne sar. *IEEE Geoscience and Remote Sensing Letters*, 12(10):2021–2025, 2015.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

Torrey, Lisa and Shavlik, Jude. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264. IGI Global, 2010.

Wang, Mingming. Multi-path convolutional neural networks for complex image classification. *arXiv preprint arXiv:1506.04701*, 2015.

Xu, Joseph Z, Lu, Wenhan, Li, Zebo, Khaitan, Pranav, and Zaytseva, Valeriya. Building damage detection in satellite imagery using convolutional neural networks. *arXiv preprint arXiv:1910.06444*, 2019.

Ying, Xue. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, pp. 022022. IOP Publishing, 2019.

Yu, Hanshun, Mohammed, Mohammed A, Mohammadi, Mohammad E, Moaveni, Babak, Barbosa, Andre R, Stavridis, Andreas, and Wood, Richard L. Structural identification of an 18-story rc building in nepal using post-earthquake ambient vibration and lidar data. *Frontiers in Built Environment*, 3:11, 2017.

Zhai, Wei and Huang, Chunlin. Fast building damage mapping using a single post-earthquake polsar image: a case study of the 2010 yushu earthquake. *Earth, Planets and Space*, 68(1):1–12, 2016.