A VARIETY OF DEEP LEARNING MODELS TO CLASSIFY DISASTER SCENE VIDEOS

[1]HAILI WANG, [2]YUAN LI, [3]BILL BUCKLES

[1,2,3]Department of Computer Science and Engineering University of North Texas, USA
E-mail: [1]hailiwang@my.unt.edu, [2]yuanli4@my.unt.edu, [3]bill.buckles@unt.edu

Abstract - The increasing frequency and severity of natural disasters has become more and more prevalent today. The fifth assessment report of the Intergovernmental Panel on Climate Change (IPCC, 2014) predicts that as global warming continues in the coming decades, its contribution to the increase in natural disaster losses will become more prominent. However, through rapid and accurate analysis of disaster scenarios, there is still an opportunity to significantly reduce catastrophic losses caused by extreme events. From a video, we extract key frames and identify embedded objects (using YOLOv3). The densely labeled images are given a global label using various VGG and ResNet tools. Classical quality measures (accuracy, precision, and recall) will guide subsequent development directions.

Keywords - Disaster Management, Deep Learning, Object Detection, Scene Classification

I. INTRODUCTION

Computer vision technologies are rapidly improving and becoming more important in disaster response. However, due to the lack of training data, many pre-existing computer vision methods cannot provide adequate support for search and rescue [3]. Fortunately, researchers at MIT have developed large-scale LADI dataset (a.k.a.low-altitude disaster image datasets) to fill the void in disaster scenario datasets [3]. We use the LADI dataset along with other open data set and open-source tools to develop deep learning models for disaster category classification from video. Specifically, our medium-term goal is to recognize multiple characteristics of video scenes that fall into main categories (flooding, landslide, fire, rubble). Furthermore, for each feature, our goal is to return a ranked list of video clips having that feature. Overall, our project aims to develop a useful and effective model that quickly responds to natural disasters by using object detection and image classification.

In section 2, we will introduce the background of image retrieval and literature review related to the topic. Section 3 focuses on the neural networks (NNs) for image classification and object recognition, including VGG, ResNet, MobileNet, and others. We detail our experiment and evaluate out model based on accuracy, precision and recall in Section4. Finally, in section 5,we conclude our findings.

II. MODEL DESCRIPTION

In recent years, the emergence of deep learning technology has revolutionized the method of target detection and greatly improved the accuracy and robustness of object detection [5].

However, for traditional scene recognition methods, the accuracy cannot meet the requirements of seismic scene recognition. There exist multiple obstacles when applying traditional scene recognition methods. Amongst these obstacles, the most prudent ones are as follows: small data sample size, the lack of experts to label the data correctly, and the complexity of disaster scenes [1][4].For disaster conditions, the amount of usable data is already confined to a limited number of data gather channels, and after filtering out the useless data, we have a meager amount of data to feed into the traditional methods. In addition, without the precise labels needed for training, the difficulty of obtaining a good accuracy increases substantially. Thus, in order to improve the accuracy of machine learning for disaster scene recognition, we will execute object detection and scene recognition on the dataset respectively.

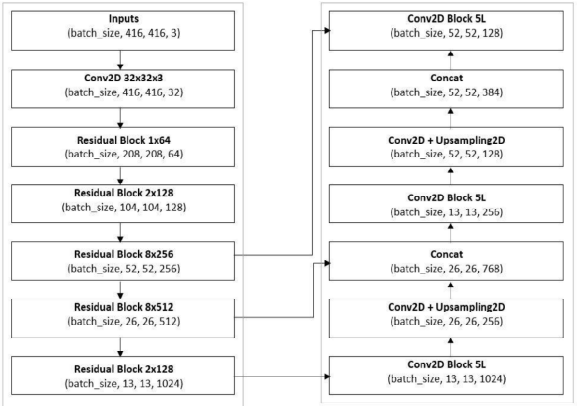


Fig. 1: Darknet-53 with FPN

2.1 Foreground Object Detection: YOLOv3

With the evolution of algorithms in the object detection field in recent years, YOLO is currently recognized as a relatively accurate object detection algorithm. The version we used in this experiment is YOLOv3. As one of the advanced real-time object detection systems, YOLOv3 [6] has both high-speed object detection and high accuracy for real-time targets. YOLOv3 applies a multi-layer NNs to the complete image, then divides the image into multiple regions and predicts the bounding box and probability

of each part. These bounding boxes are weighted by the prediction probability. The structure and design of it allows its application to foreground object detection.

YOLOv3 uses a fully convolutional network composed of residual blocks as the backbone network, with a network depth of 53 layers, named Darknet-53 by the authors of YOLOv3. Figure 1, part A, shows the detailed structure of Darknet-53. YOLOv3 draws on the idea of feature pyramid network (FPN) and extracts features from different scales. In contrast to YOLOv2, which only extracts features in the last two layers, YOLOv3 expands the scale to the last three layers. Figure 1, part B, is based on Part A with an illustration of the multi-scale feature extraction part.

YOLOv3 does not use softmax to classify each box, but uses multiple logistic classifiers, because softmax is not suitable for multi-label classification, and the accuracy of independent multiple logistic classifiers will not decrease.

2.2 Background Classification

1) VGGNet: Very Deep Convolutional Networks [7](VGG) use three 3x3 convolution kernels instead of 7x7 convolution kernels and two 3x3 convolution kernels instead of 5x5 kernels. Under the same perception field, the depth of the network is improved, and the effect of the NN is improved to a certain level. The figure2 shows the structure of VGG Networks including VGG16 and VGG19 [7]. Specifically, VGG16 contains 16 hidden layers including13 convolutional layers and 3 full connection layers as shown in column D in the figure, while VGG19 contains 19 hidden layers including 16 convolutional layers and 3 full connection layers as shown in column E in the figure. The structure of VGG network is very consistent, and We use VGG16 and VGG19 to classify the test images into 'whole scene' categories.

2) ResNet: The deep residual learning network-ResNet is designed to solve the degradation problem.

As shown in Figure 3, the method is to make these layers fit residual mapping, rather than make each stacked layer fit the desired underlying mapping directly. Assuming that the desired underlying mapping is H(x), let the stacked nonlinear layerfit the other mapping: F(x):= H(x) - x. Therefore, the original mapping is going to be F(x) + x, which means residual mapping is easier to optimize than the original unreferenced mapping. The formula F(x) + x can be implemented through the "shortcut connection" of neural network in which one or more layers are skipped [2].Residual networks (ResNet) have simple structures which can solve the problem of deep CNN performance degradation under extremely deep conditions. They also have excellent classification performance. Widespread use of residual networks has pushed the performance of computer vision tasks to new heights. We are training the ResNet model (ResNet50 and ResNet101) to classify the filtered test images to analyze the accuracy and compare the results with the other neural networks such as VGG16 and VGG19.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 2: VGG network structure

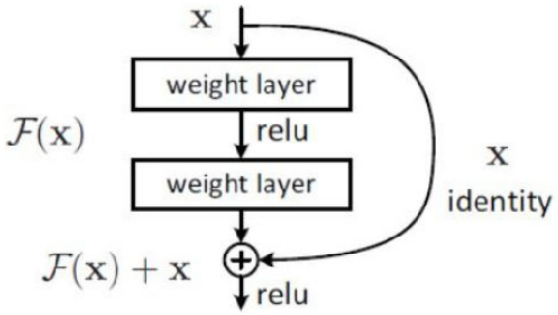


Fig. 3: Residual Learning: a Building Block

III. EXPERIMENT AND ANALYSIS

The test data set – LADI dataset contains41 original full videos and 1,825 segmented short video clips between 2 to 20 seconds. First, we prepare the segmented videoclips and extract key-frames that contain the information in multiple scenes. Next, we filtered out the minimum number of frames that can represent a specific scenario. With the data cleaning and pre-processing stages complete, we then utilize YOLOv3 for foreground object detection and filter out useful images to the experimental data. Finally, we use the above models for background classification. According to our data set, we classify all experimental images into three major categories of disasters, which are flooding, damage, and landslide.

3.1. Data Processing: Key Frames Extraction

During the key frames extraction process, we use local maxima as the final step. From a video, the inter-frame differences are computed. Using local

maximum, the frames for which the average inter-frame difference are local maxima are selected as keyframes. The extraction results obtained via this method perform better in diversity, and the extraction results are evenly dispersed within the video.



Fig. 4: YOLOv3 Result Shot3 002 246

3.2. Key Frames Filter

Upon extracting key frames for each video clip, the sub- data contains repetitive key frames for the same scene. At this point, we screen 1-2 frames per scene to reduce the data volume. From Figure 5, we note the following discoveries:

1) With the pre-processing step, the raw data will be presented by sequences of key frames images. 2) shot2- 000-163, will be saved and used as the only valid image representing shot 2 number 000 video clip. By completing the above steps, 1,885 images are selected for future use, so that we can successfully minimize the size of raw test data to improve classification efficiency of our model. Figure 5presents the example test result for this step, shot2-000-163, is cached for the background classification step.



(a)Shot2-000-53 (b)Shot2-000-111



(c)Shot2-000-163(d)Shot2-000-265

Fig. 5: Key Frames Extraction Result for Shot2-000

3.3. Foreground Object Detection

Since YOLOv3 has great advantages in detecting small objects, in this step, we used YOLOv3 to process the image dataset which we generated from the previous data processing step, based on our overall evaluation of the data and the detection of foreground objects in the subsequent part, we found that for our data in three major natural disasters: damage, flooding, and landslide, we focus more on vehicle detection. According to the comparison of the existing excellent databases, we found that the PASCAL VOC and COCO datasets pertaining models can use in our experiment.

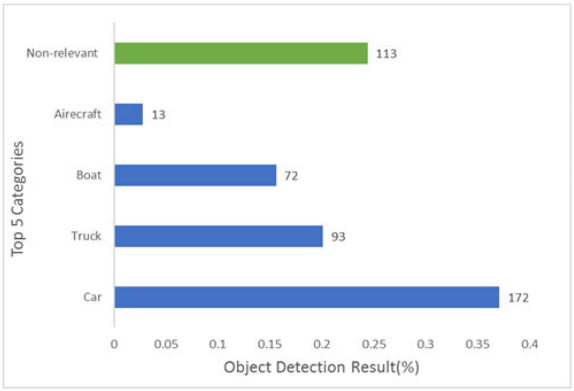


Fig. 6: YOLOv3 Object Detection Results

3.4. Background Classification

We finally identified 370 images for background classification, of which 15% for testing and 85% for training. The distribution of experimental data is shown in the figure 7. Also, we use TensorFlow framework flip function: horizontal, random horizontal, vertical, random vertical, rotation range, width shift range, height shift range, zoom range, rescale, and rotation range to enlarge our sub- dataset. The background classification model we will use are presents in the previous section. In this part, we will train each model and then test it accordingly.

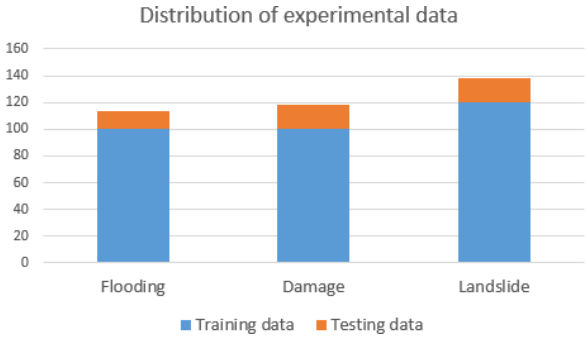


Fig. 7: Distribution Experimental Data

IV. RESULT

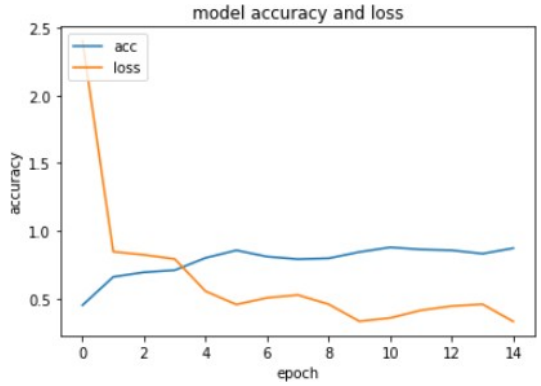
From our experimental results, we find that: we used YOLOv3 to process 1,885 image dataset which generate from the data processing stage and finally 350 images containing target objects were obtained. The object detection result is shown in figure 6. Also stemming from our results, we observe YOLOv3's successful extraction of targets such as vehicles, boats, and airplanes. Figure 4 presents the object detection result from YOLOv3 which detects cars and trucks.

In the following experiment, we use the TensorFlow framework to substantive evaluate the performance of each model. Based on the previous step, we originally planned to perform further background classification on the 350 images which selected by YOLOv3, but due to the limitations of the data, these images cannot completely contain all three types of disasters required for the experiment. To obtain a more comprehensive experimental data, in addition to the pictures selected by YOLOv3, we also selected images of different disaster types and added them to the experiment. Finally, we identified 370 images for background classification, of which 15% for testing and 85% for training. The distribution of experimental data is shown in figure7.

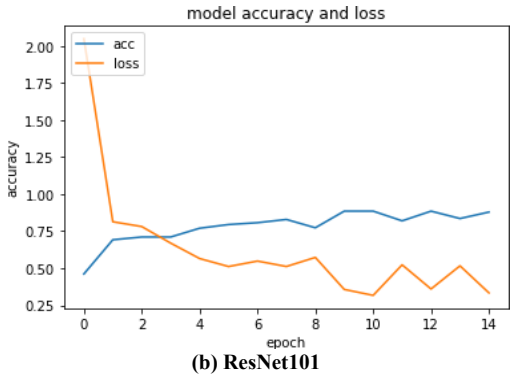
To measure the proportion of different experimental models, we use accuracy, precision, recall, f1-score and confusion matrix as metrics.

VGG16			
Disasters	**Precision**	**Recall**	**F1-score**
Damage	**0.76**	**0.72**	**0.74**
Flooding	**0.78**	**0.29**	**0.42**
Landslide	**0.51**	**0.80**	**0.62**
VGG19			
Disasters	**Precision**	**Recall**	**F1-score**
Damage	**0.82**	**0.72**	**0.77**
Flooding	**0.66**	**0.79**	**0.72**
Landslide	**0.83**	**0.83**	**0.83**

TABLE I: VGG16 & VGG19 Results



(a) ResNet50



(b) ResNet101

Fig. 8: Accuracy and Loss of ResNet50 vs ResNet101

A. VGGNet

Table I describes the results of the two models of VGG. By comparing precision, we can conclude that in the comprehensive comparison of the results of the three disasters, VGG19 has better experimental results than VGG16, and can achieve the precision of 83% of the landslide. In fact, compared with VGG16, the accuracy of VGG19 in predicting these three disaster scenarios is greatly improved.

B. ResNet

Figure 8 presents the accuracy and loss curve of ResNet50 and ResNet101. According to the graph, we can conclude that the performance of accuracy and loss curve is very similar for both models. However, there are some minor differences between the curves of the different models. For example, under epoch 10, when ResNet101 reach its lowest point of loss, while ResNet50 needs more epochs to train. And for the accuracy of the experimental results, the accuracy of ResNet101 is still increasing slowly, while ResNet50 does not show an increasing trend after epoch 6.

Table II presents the results of Resnet50 and Resnet101 on our testing data. By comparing these two results, Resnet50 has a higher accuracy in predicting flooding and landslide scenarios, while Resnet101 has higher accuracy in predicting damage images. Overall, Resnet50 seems has a better performance than Resnet101. Although these differences exist under the current conditions, we predict that the results of the two models will close the gap as the volume of test data increases.

ResNet50			
Disasters	**Precision**	**Recall**	**F1-score**
Damage	**0.67**	**0.92**	**0.77**
Flooding	**0.86**	**0.92**	**0.89**
Landslide	**0.94**	**0.71**	**0.81**
ResNet101			
Disasters	**Precision**	**Recall**	**F1-score**
Damage	**0.94**	**0.68**	**0.79**
Flooding	**0.64**	**0.9**	**0.75**
Landslide	**0.66**	**0.71**	**0.69**

TABLE II: ResNet50 & ResNet101 Results



```
loading image Shot24_045_160.png
[[0.00074431 0.5895501  0.4097056 ]]
The predicted type of img is:  1
```

Fig. 9: An Example of Wrong Predictions

Figure 9 shows an example of ResNet101 performing a false prediction. We classify flooding, damage, and landslide into categories 1, 2 and 3. The three numbers in brackets represent the percentage of each disaster. For example, taking the results from 9, with the damage of 0.589 and the landslide of 0.409 indicates that the image displayed that the image displayed had a 58% chance of being classified as a damage scenario when the image should be classified as a landslide.

V. CONCLUSION AND FUTURE WORK

In this paper, we implemented the extraction of keyframes in the video format of the raw LADI dataset, and successfully performed the foreground object recognition based on the YOLOv3 framework with the extracted images dataset to form a new sub-data set. A variety of neural networks, including VGG, ResNet, and MobileNet, were used to train the reorganized sub-data set and to obtain comparative results. In our experiment, we use an existing neural network to process the sub-data set. Existing neural networks are not common for disaster image processing, so our experimental results cannot be compared with other tasks under the same network type. Due to the limitation of the YOLOv3 pre-training data set, more improvements are needed for foreground object recognition. Our work can be further improved by identifying foreground objects from a more detailed perspective, as well as analyze the error results of recognition, in order to improve the results of foreground object detection. For background classification, the processing speed and accuracy of scene recognition can be the primary goal of optimization in the next step.

REFERENCES

[1] N. Chaudhuri and I. Bose. Application of image data analytics for immediate disaster response. In Proceedings of the 21st International Conference on Distributed Computing and Networking, pages 1–5, 2020.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer visionandpatternrecognition,pages770–778,2016.

[3] J. Liu, D. Strohschein, S. Samsi, and A. Weinert. Large scale organization and inference of an imagery dataset for public safety. In 2019 IEEE High Performance Extreme Computing Conference (HPEC), pages 1–6. IEEE,2019.

[4] J. Mao, K. Harris, N.-R. Chang, C. Pennell, and Y.Ren.Train and deploy an image classifier for disaster response. arXiv preprint arXiv:2005.05495,2020.

[5] V. Nunavath and M. Goodwin. The role of artificial intelligence in social media big data analytics for disaster management-initial results of a systematic literature review. In 2018 5th International Conference on Information and Communication Technologies for Disaster Management(ICT-DM),pages1–4.IEEE,2018.

[6] J. Redmon and A. Farhadi. Yolov3: An incremental improvement.

[7] arXiv preprint arXiv:1804.02767, 2018.

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556,2014.

★★★