Integrating Multiple Deep Learning Models to Classify Disaster Scene Videos

Yuan Li*, Haili Wang*, Shuo Sun*, and Bill Buckles†
Department of Computer Science and Engineering
University of North Texas, USA
*{yuanli4, hailiwang, shuosun2}@my.unt.edu, †{bill.buckles}@unt.edu

***Abstract*—It is well-known that the trend of increasing global natural disasters of natural disasters globally, accompanied by increasing loss of life and property, shows no signs of halting. 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, ResNet, and MobileNet tools. Classical quality measures (accuracy, precision, and recall) will guide subsequent development directions.**

***Index Terms*—Disaster management , Deep learning, Object detection, Scenes classification**

I. Introduction

It is well-known that the trend of increasing global natural disasters of natural disasters globally, accompanied by increasing loss of life and property, shows no signs of halting. The fifth assessment report of the Intergovernmental Panel on Climate Change (IPCC, 2014) predicts that as global warming continues, its contribution to the increase in natural disaster losses will become more prominent [6]. There is no sensible way to prevent the occurrence of natural disasters currently. However, through rapid and accurate analysis of disaster scenarios, there is still an opportunity to significantly reduce catastrophic losses caused by extreme events.

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 [9]. 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 [9]. We use the LADI dataset along with other open data set and open source tools, in order 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 Section 4. Finally, in section 5, we conclude our findings.

II. Background

As the main source of information for processing natural disasters cannot be easily discerned from the raw data such as video footage, we look at alternate methods of information extraction. A survey by Alkhawlani, et. al [1] proposes the taxonomy in Fig. 1. Seldom does a specific image retrieval system fall neatly into one category. For example, both Yahoo Image Search and Google Image Search are keyword-based but both allow filtering based on parameters such as image size, image format, color scheme, and other metadata (i.e., 'Descriptors' in the figure). Additionally, both systems provide user images that are similar to each one directly matched (i.e., 'Association' in the figure).

Text-based image retrieval systems have obvious disadvantages. Manual labeling is required and the labels are expressed in one specific natural language. Content-based image retrieval (CBIR) extracts off-line descriptors from exemplar images and compares them for similarity with the same measures extracted from images of unknown category. Local features are exemplified by keypoints such as SIFT [10], by texture [17], by color, using HoG (histogram of gradients), and by shape. Global features are exemplified by color histograms or low-pass coefficients from a transform. Most often, one local feature is not used in isolation. Rather several are combined via a vector or a Bag-of-Features model [16].

Note that color and HoG are particular extendable to an image as a whole. A color histogram or histogram of gradients can be extended to encompass the entire image. By vectorizing, most local features can be expanded to provide a whole-image description.

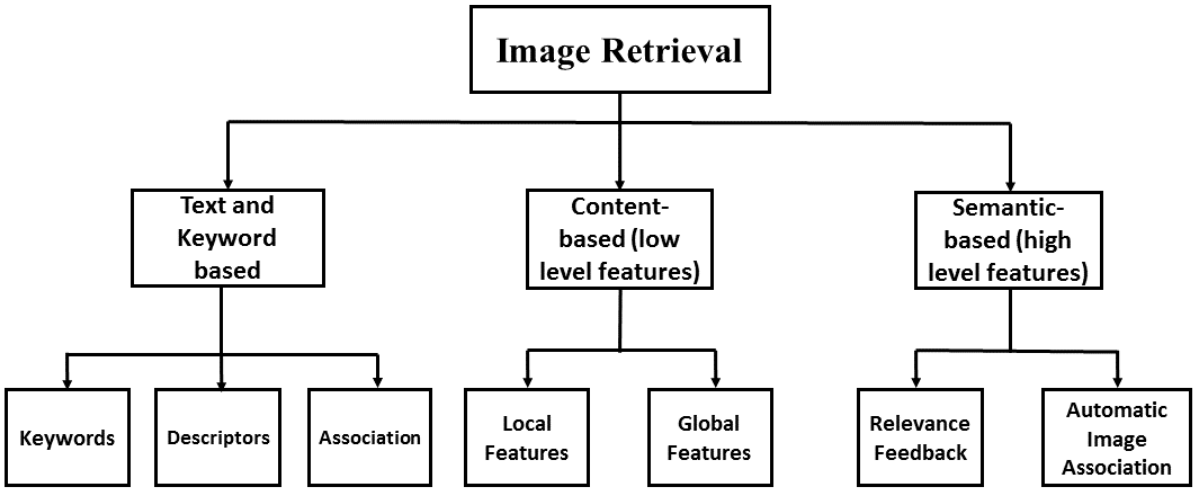


Fig. 1: Image Retrieval Taxonomy [1]

Whether local or global features are extracted, a method must exist to compare the descriptor of two images. Commonly a descriptor is represented as a structured vector – a vector for which one or more components are structured vectors. A different metric might be assessed for each component. Metrics applicable include Euclidean distance, KL-divergence [3], fuzzy logic [4][19][20], rough sets [14], and others. CBIR systems have the disadvantage that all measures are ultimately founded upon the lowest level feature – the color of a pixel. The human vision system perceives from a different basis. This is known as the 'semantics gap'.

Deep learning has been a factor in image captioning and CBIR for just over a decade. Already there are a number of excellent surveys [7][18][21]. When deep learning is implemented as convolutional neural nets (CNNs), it may be characterized as aggregating multiple local features into a global descriptor ranked against a set of exemplars. One researcher [18] claims a pre-trained CNN outperforms current feature extraction approaches.

III. 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 [12]. However, for traditional scene recognition methods, the accuracy cannot meet the requirements of seismic scene recognition. There exists 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 [2] [11]. 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 data set respectively.

A. 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 [13] has both high-speed object detection and high accuracy for real-time targets. YOLOv3 applies a multi-layer NNs to the complete complete image, then divides divides the image into multiple regions and predicts the bounding box and probability of each part. These bounding boxes are weighted

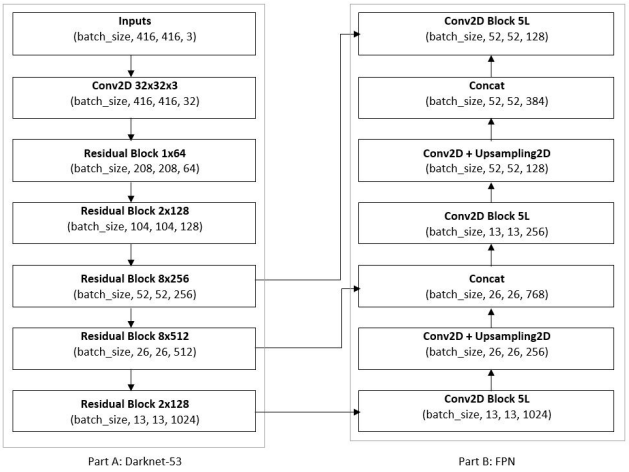


Fig. 2: Darknet-53 with FPN

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. 3: VGG network structure

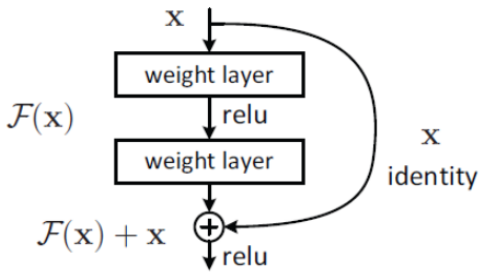


Fig. 4: Residual Learning: a Building Block

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.

B. Background Classification

1) VGGNet: Very Deep Convolutional Networks [15] (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 figure3 shows the structure of VGG Networks including VGG16 and VGG19 [15]. Specifically, VGG16 contains 16 hidden layers including 13 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 4, 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 $\mathcal{H}(x)$, let the stacked nonlinear layer fit the other mapping: $\mathcal{F}(x) := \mathcal{H}(x) - x$. Therefore, the original mapping is going to be $\mathcal{F}(x) + x$, which means residual mapping is easier to optimize than the original unreferenced mapping. The formula $\mathcal{F}(x) + x$ can be implemented through the "shortcut connection" of neural network in which one or more layers are skipped [5]. Residual networks (ResNet) have simple structures that 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 in order to analyze the accuracy and compare the results with other NNs.

3) MobileNetV3: The Figure 5 above show the network block structure of MobileNetV2 and MobileNetV3. The MobileNetV3 model is a combination of the following three ideas: MobileNetV1's depth-wise separable convolutions, MobileNetV2's inverted residual with linear bottleneck, and MobileNet's squeeze and excitation structure based on lightweight attention model [8]. Combining the advantages of the above three structures, an efficient MobileNetV3

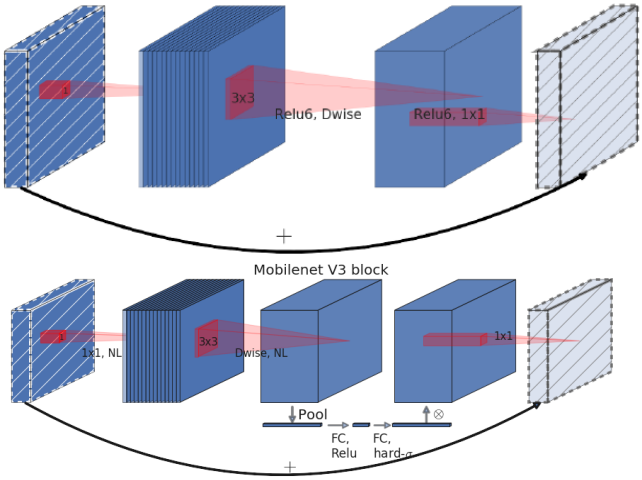


Fig. 5: MobileNetV2 Layer vs MobileNetV3 Block

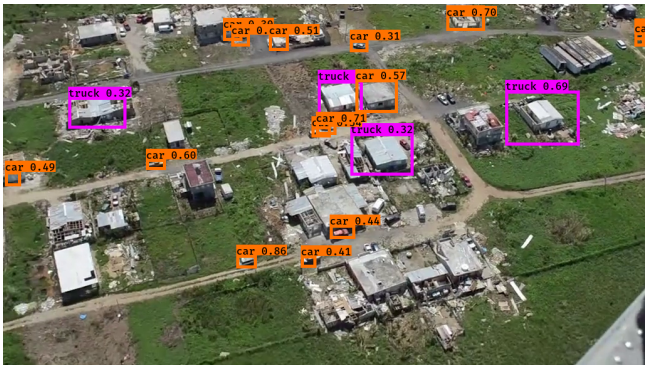


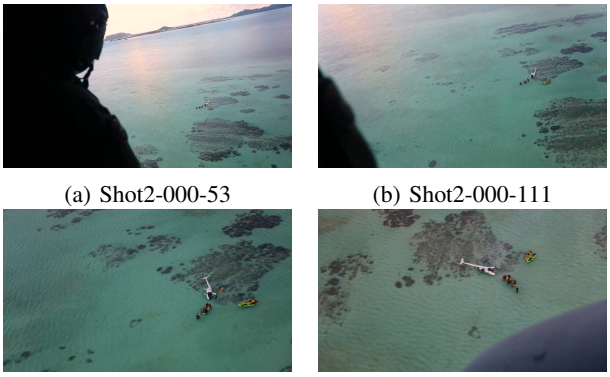
Fig. 6: YOLOv3 Result_Shot3_002_246

module is designed.

Overall, MobileNetV3 has two innovation points. First, it combines search technologies, with modular search performed by hardware-aware network architecture search (NAS) and local search performed by NetAdapt. In addition, MobileNetV3 improves the network structure by introduce the h-swish activation function[8].

IV. Experiment and Analysis

The test data set – LADI dataset contain 41 original full videos and 1,825 segmented short video clips between 2 to 20 seconds. First, we prepare the segmented video clips and extract key-frames that contain the information in multiple scenes. Next, we manually filter 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.



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

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

Fig. 7: Key Frames Extraction Result for Shot2-000

A. 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 key frames. The extraction results obtained via this method perform better in diversity, and the extraction results are evenly dispersed within the video.

B. 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 manually screen 1-2 frames per scene to reduce the data volume. From Figure 7, 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 7 presents the example test result for this step, shot2-000-163, is cached for the background classification step.

C. 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.

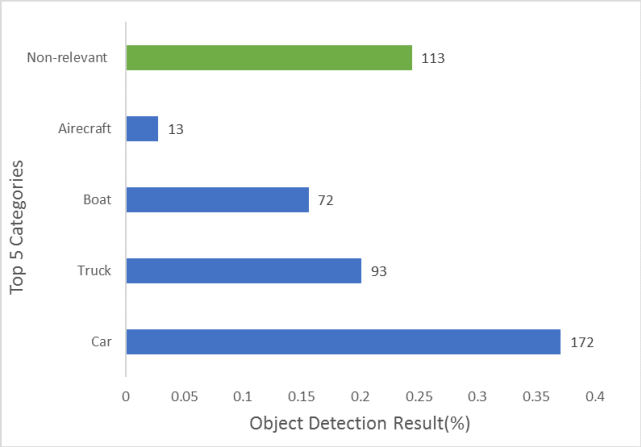


Fig. 8: YOLOv3 Object Detection Results

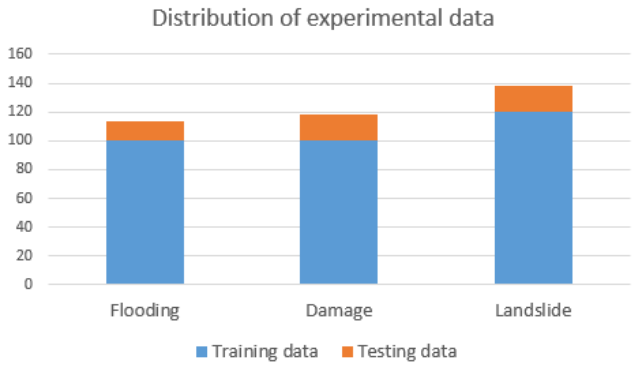


Fig. 9: Distribution Experimental Data

D. 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 figure 9. 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 subdataset. The background classification model we will use are presents in the previous section. In this part, we will train each model and test it accordingly.

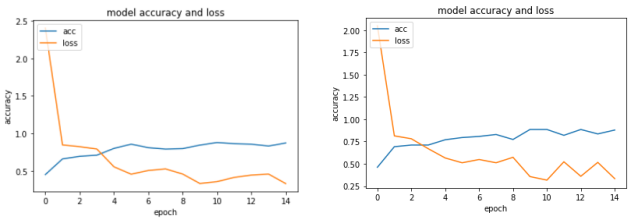
V. Result

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 8. Also stemming from our results, we observe YOLOv3's successful extraction of targets such as vehicles, boats and airplanes,etc. Figure 6 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

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. 10: Accuracy and Loss of ResNet50 vs ResNet101

models. 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. In order to have more comprehensive experimental data, in addition to the pictures selected by YOLOv3, we also manually 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 figure 9.

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

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. Specifically, compared with VGG16, the accuracy of VGG19 in predicting these three disaster scenarios is greatly improved.

B. ResNet

Figure 10 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,

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

Fig. 11: An Example of Wrong Predictions

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.

Figure 11 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 11, 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 actually be classified as a landslide.

C. MobileNet

Figure 12 presents the accuracy and loss curve of MobileNetV2 and MobileNetV3. While comparing the loss curves, we can find that the loss reductions of the two models are roughly similar. After epoch 8, the loss curve stabilizes. Comparing the accuracy of the two models, we can find that although the accuracy of the two models is relatively close at epoch 20, the accuracy of MobileNetV2

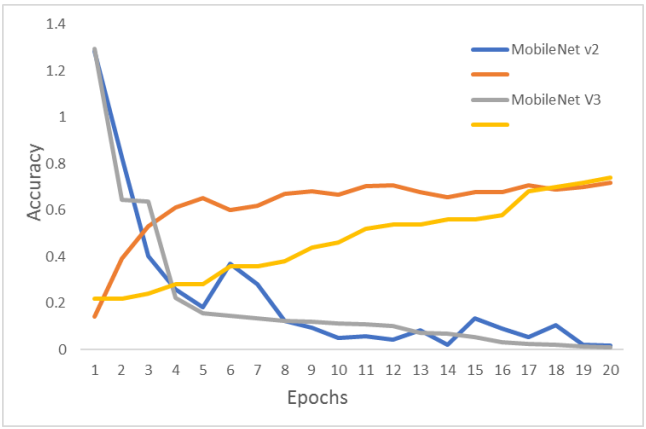


Fig. 12: Accuracy and Loss of MobileNetV2 vs MobileNetV3

has not improved significantly after epoch 5. Nevertheless, the accuracy of MobileNetV3 is still in a stage of continuous growth.

VI. Conclusion and Future Work

In this paper, we implemented the extraction of key frames 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] M. Alkhawlani, M. Elmogy, and H. El-Bakry. Text-based, content-based, and semantic-based image retrievals: A survey. *International Journal of Computer and Information Technology*, 4:58–66, 01 2015.

[2] 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.

[3] S. Cui and M. Datcu. Comparison of Kullback-Leibler divergence approximation methods between Gaussian mixture models for satellite image retrieval. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3719–3722, 2015.

[4] S. Dhingra and P. Bansal. A competent and novel approach for designing image retrieval systems. *EAI Trans. on Scalable Information Systems*, pages 1–13, 2019.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] P. Hoeppe. Trends in weather related disasters–consequences for insurers and society. *Weather and climate extremes*, 11:70–79, 2016.

[7] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys*, 51(6):1–36, 2019.

[8] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1314–1324, 2019.

[9] 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.

[10] D. G. Lowe. Distinctive image features from scale invariant keypoints. *Intern. J. of Computer Vision*, 60(2):91–110, 2004.

[11] 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.

[12] 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)*, pages 1–4. IEEE, 2018.

[13] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[14] M. Shahabi Lotfabadi, Y. Zhan, and A. Bashirzadeh Tabrizi. Evaluating a cover based rough set classifier in a content based image retrieval system. In *International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, pages 122–129, 2018.

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

[16] A. Suharjito and D. D. Santika. Content based image retrieval using bag of visual words and multiclass support vector machine. *ICIC Express Letters*, 11:1479–1488, 10 2017.

[17] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visiual perception. *IEEE Trans. on Systems, Man and Cybernetics*, 8(6):460–473, June 1978.

[18] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM International Conference on Multimedia*, page 157–166, New York, NY, USA, 2014. Association for Computing Machinery.

[19] V. Yadaiah, R. Vivekanandam, and R. Jotothu. A fuzzy logic based soft computing approach in cbir systems using incremental filtering feature selection to identify patterns. *Int. J. of Applied Engineering Research*, 13(5):2432–2442, 2018.

[20] R. R. Yager and F. E. Petry. A framework for linguistic relevance feedback in content-based image retrieval using fuzzy logic. *Information Sciences*, 173:337–352, 2005.

[21] L. Zheng, Y. Yang, and Q. Tian. SIFT meets CNN: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1224–1244, 2018.