# Independent Category Classifiers for Emergency Scene Description using Deep Learning approaches

Mirko Zaffaroni[1,2], Federico Oldani[1], and Claudio Rossi[1]

[1]LINKS Foundation, Turin, Italy
[2]Department of Computer Science, University of Turin

## Abstract

In this work, we present our proposed model for Disaster Scene Description and Indexing (DSDI) Challenge of TRECVIDI2020. For the challenge we used: the LADI Dataset, a dataset composed of scrapped Google Images using as a keyword the name of the label and an extended set of the LADI dataset. This extension was created using a crowdsourcing service like Amazon Mechanical Turk. As approaches we tried different combination of Convolutional Neural Networks (CNN), what worked better for us was using five different classifiers, one for each category of the LADI dataset. We used this configuration because we noticed that dividing the task lead to better scores. Indeed by checking the results it is possible to notice that dividing the task help the model to learn specific features for that category. We found that the dataset is very challenging and it is difficult for a model trained end to end to learn all the features useful to detect a class. For this reason, an ensemble model approach worked better for the challenge. We think that more sophisticated label for example segmentation map could have allowed obtaining better results.

## 1 Introduction

The TRECVID2020 [1] Disaster Scene Description and Indexing (DSDI) Challenge involves using a newly developed dataset to solve a multi-labelling task. The Task consists of detecting all the label that were detected as feasible for an emergency scenario. Labels can be grouped into five macro-categories: *Damage, Environment, Infrastructure, Vehicles and Water*; each of these categories is composed of different elements, each of these distinguishes a concept or an object. The peculiarity of this challenge lies in the fact that not all the classes that must be predicted consist of entities, as *infrastructure* and *vehicle* others concern more extensive elements that can also cover all the area of the image, like the elements of the *environment* or *water* categories, finally in *damage* there are mostly conceptual classes that are difficult to define at the local level (it is difficult to define a bounding box that enveloped all the interesting area) and can be evaluated by a person only by analyzing the image in its entirety, these conceptual labels are also the most difficult to predict because often their set of features can be very heterogeneous. These are the challenges that this dataset poses, but if they can be overcome, the dataset can become an important element in classifying and analyzing images of natural disasters and can become a useful tool for first responders. This Challenge is, therefore, the first step to gain a better understanding of the dataset and to understand what works must be carried out in this direction to improve the first intervention in emergency scenarios thanks to open data now available online.

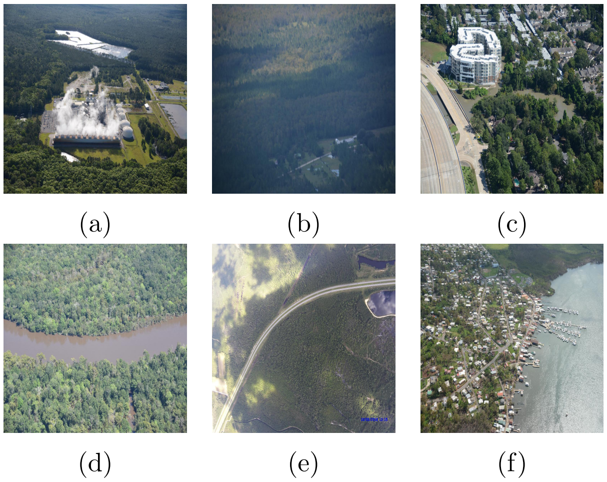

(a) (b) (c) (d) (e) (f)

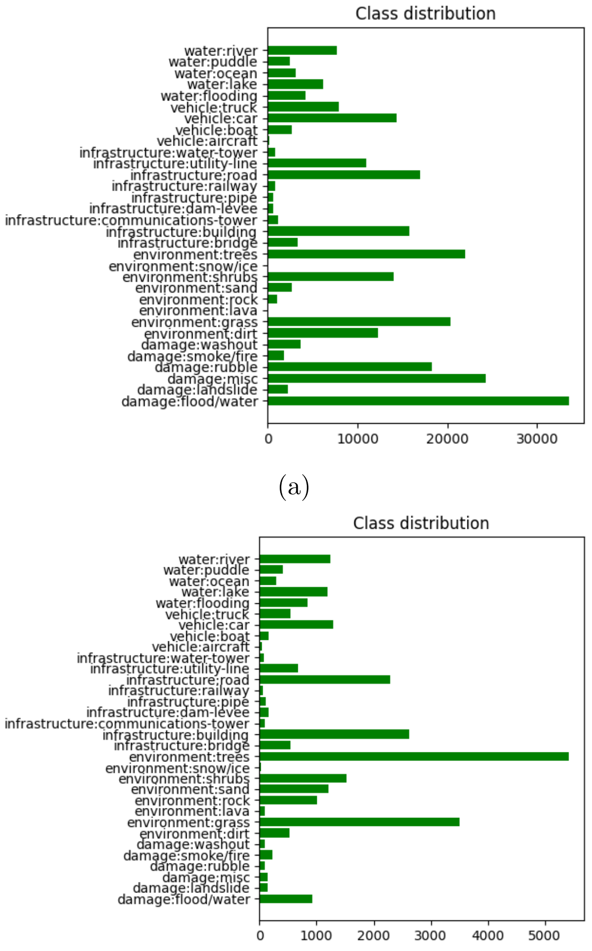
Figure 1: Example of images contained in the dataset. It can be noted that the lighting, orientation, perspective, and resolution varies across the examples. These changes are a key component to the LADI Dataset and are part of the main challenges that this dataset has to offer.

## 2 LADI Dataset

The dataset developed for the challenge is composed of a set of images gathered by Civil Air Patrol during missions for various natural disasters. It presents various challenges due to the angle from which the images were collected. The change in the angulation is a problem because if it differs too much from the one used to train state of the art neural network there is a great chance they will perform poorly. In Fact, usually, the images, which they are trained on, are collected from land and not from an aerial point of view. Furthermore, the type of label of the dataset is not homogeneous. The data are divided into entities, environmental elements and concepts. This represents another of the challenges for this dataset. In fact, if some entities (house, road, bridge, truck, car, etc ...) can be easily described with features, it is slightly complicated for environmental elements, that can extend over the whole image (tree, river, ocean, etc ...), to be identified, since, being background elements, they can contain within them entities that are still part of the set of features for the environmental element. While in the *damage* category there are all elements of damage to structures or other elements, however, the damage is something conceptual, for this reason even a person would find it difficult to classify and define it with a set of features. This type of label is the most difficult to classify within the dataset. Here in Figure 1 are presented some examples of images of the dataset.

While in the Figure 2 the distributions of the dataset are presented, in the first image 2a the distributions of the dataset provided for the challenge are presented, while in the second image 2b the distributions for the extended dataset we have collected are represented by taking a non-labelled subset of the LADI Dataset and having it labelled using Amazon Mechanical Turk as a service.

As can be seen from Figure 2 the distributions for the dataset and its extension are very unbalanced on the classes, some are very predominant



(a)

(b)

Figure 2: The different class distributions for the LADI Dataset and for the dataset extension we propose using Amazon Mechanical Turk as a labelling service. As you can see from the images the classes are very unbalanced and for some classes, the examples are very few.

(trees, grass, roads), while others have very few examples (lava, snow/ice, aircraft). For this reason, given the scarcity of examples for certain classes, it will be really difficult to get good detections for these classes that have very few examples.

# 3 Models

As model we used ResNet152 [2] as feature extractor, ResNet allows to extract pretrained features trained on Imagenet [3]. The features learned from the model with the pretraining on Imagenet are mostly features of images taken from the ground and some of these features, although very similar to the aerial ones, in some cases, they may be different. For example, a tree can have similar leaf features when viewed from above, but a house or any other building can change a lot. A first vanilla model, represented in Figure 3, consisted in trying to use a single backbone network to extract the features of all the classes and then passing these features to a classifier capable of identifying the presence or absence of a certain class. However, this approach was the one that obtained the worst results in the validation phase, most likely the complexity of the dataset is so high and the images provided in training although numerous 30k were not enough to ensure correct learning of all the features of the dataset. The model used was Resnet152, a fairly complex model, able to learn features for the 1000 classes of Imagenet, so if it fails to learn the correct features for the LADI Dataset this is most likely due to the need to have a much larger dataset. The other proposed model, in 4, consists of a single feature extractor, and five different classifiers, one per category. This model under validation was able to achieve better results. Finally, as the last step, we decided to try a separate approach for each category and finally combine the results. This model, which is represented in Figure 5, is the one that obtained the best results in the validation phase and is the one we have selected as the final model. In the experiments section, the implementation choices and the experiments carried out using these models will be presented in more detail.

## 4 Experiments

We tested different classifier configurations before proposing the optimal variant for the challenge. A first experiment was performed using a single network as a features extractor, in our case we used ResNet152, followed by a classifier, a simple multi-perceptron network. This model, even in the testing phase, has given very unsatisfactory results, but we wanted to try it anyway to evaluate the complexity of the task and have a starting point from which to advance our work. Following these results, we thought that maybe the problem was in the classifier, which is too simple and not able to correctly find a division between classes. For this reason, we have decided to separate the classifier into 5 distinct networks, one for each category. In fact, the categories although they are very heterogeneous among them, within the same category the classes are very similar and therefore with this configuration, we have tried to make sure that each classifier is particularly specific for the features of that category. The experimental results showed that this approach was better than the previous one since the scores obtained were improved. These results prompted us to try a further split of the model, also dividing the feature extraction part, dedicating one for each category. The idea was very similar to that of the previous case if the model benefits from having a dedicated classifier for each feature, perhaps it could also benefit from having more specific features learned for that category, this would lead the classifier to have the possibility to further specialize for that specific set of features of the category. The experimental results showed that this configuration based on backbones and separate classifiers was the best, as the scores improved further. Having defined the best model, we wanted to try some variations in the dataset. A first variation consisted of extending the dataset with an additional 6k of examples. To obtain this data we used the Amazon Mechanical Turk platform [4], on a random set of examples that had not already been labelled. We have seen that an improvement can be achieved by extending the dataset. This suggests that given the complexity of the task, the 35k examples provided by the challenge are not enough and therefore that the algorithms would certainly benefit from a greater amount of data. This observation is not only valid for this case, but it has been shown several times that the performance of convolutional networks is also often correlated with the number of examples available during training. Another experiment that we tried, which did not produce great benefits, but we report for the record and experimental interest, consisted in carrying out a pretraining phase using images taken by Google. The images were collected using an automatic system based on a keyword query and then download the first 1000 images per keyword. This was done using the label name as a keyword. After downloading the 32k of images (1k for each class) the model backbone was pretrained on a classification task on the 32 classes of the LADI Dataset, in this way the network should have had a pre-training of the features of the target dataset. However, even with these images the problem of perspective persists, since almost all

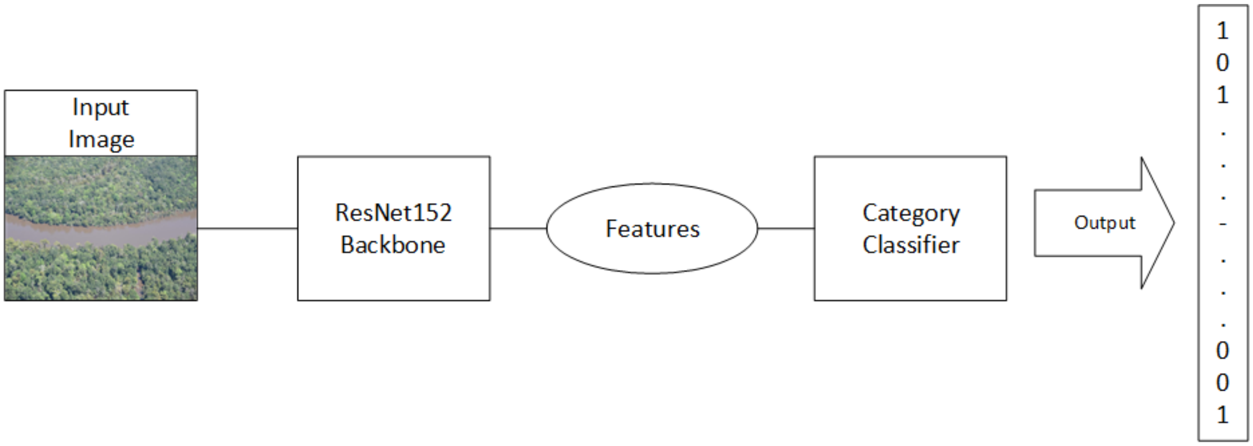


Figure 3: Single classifier

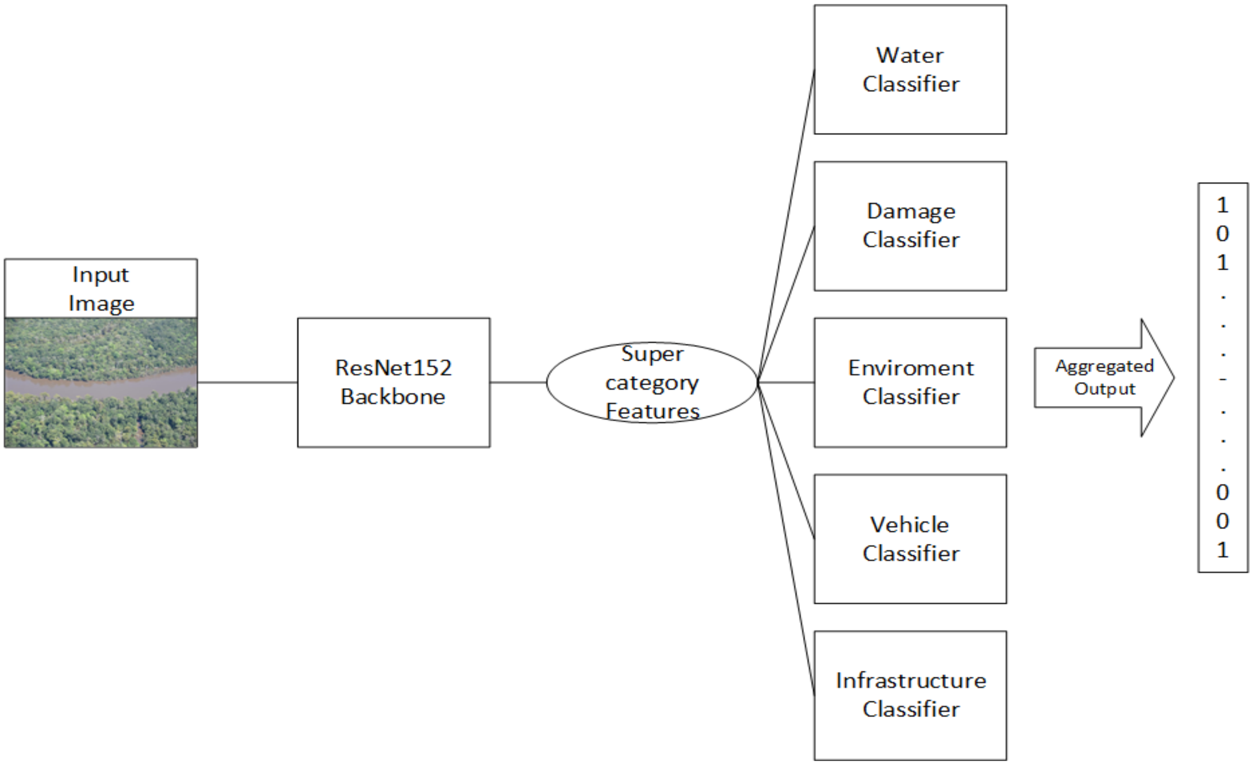


Figure 4: Five classifiers

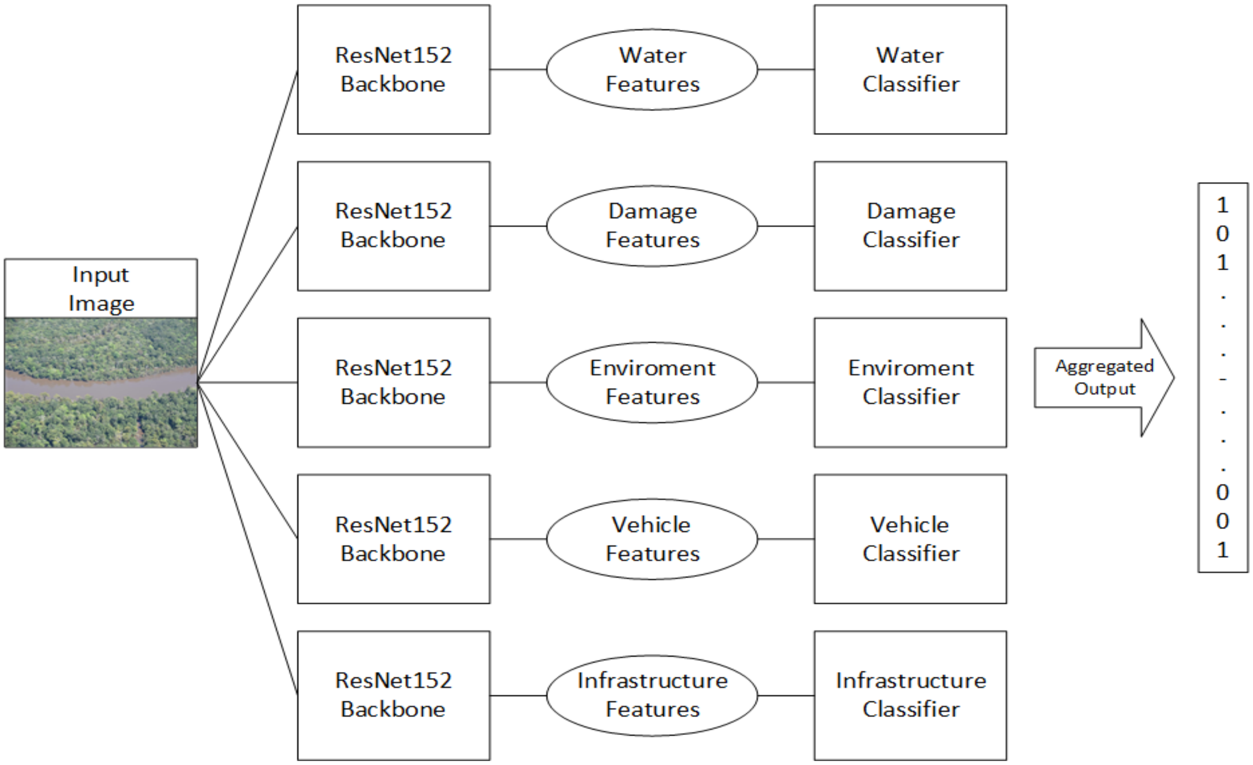


Figure 5: Five Networks

the images had a perspective from the ground and not from the air-view. Following this pretraining, the backbone network was reinserted into the five separate network model to see if this pretraining phase had given any benefit in identifying better features for the task. However, from the experimental results, this process seems to not influence the final scores. In Table 1 it is possible to see the experimental results obtained on the challenge test set. Figure 6 instead shows the scores obtained by each team in the challenge, our team results for each run are highlighted in red. For each run the training approach was the same, the only difference was the aggregation mode of the label for the test set and the last run that was based on a pretraining on Google Images. From left to right the runs are the following: *1)*: 1000 video sequences, aggregated by max correspondence in the sequence; *2)*: sequences were aggregated with *mean* but filtered out with a threshold t=0.25; *3)*: the same as the previous run, but with *max* as aggregation function and t=0.5; *4)*: done with a pretrain on Google Images and by using *max* and the first 1000 sequences.

We perform both the training and the testing on a desktop workstation with an Intel(R) Core(TM) i9-7940X CPU, 128 GB RAM and 4 NVIDIA GeForce GTX 1080 Ti.

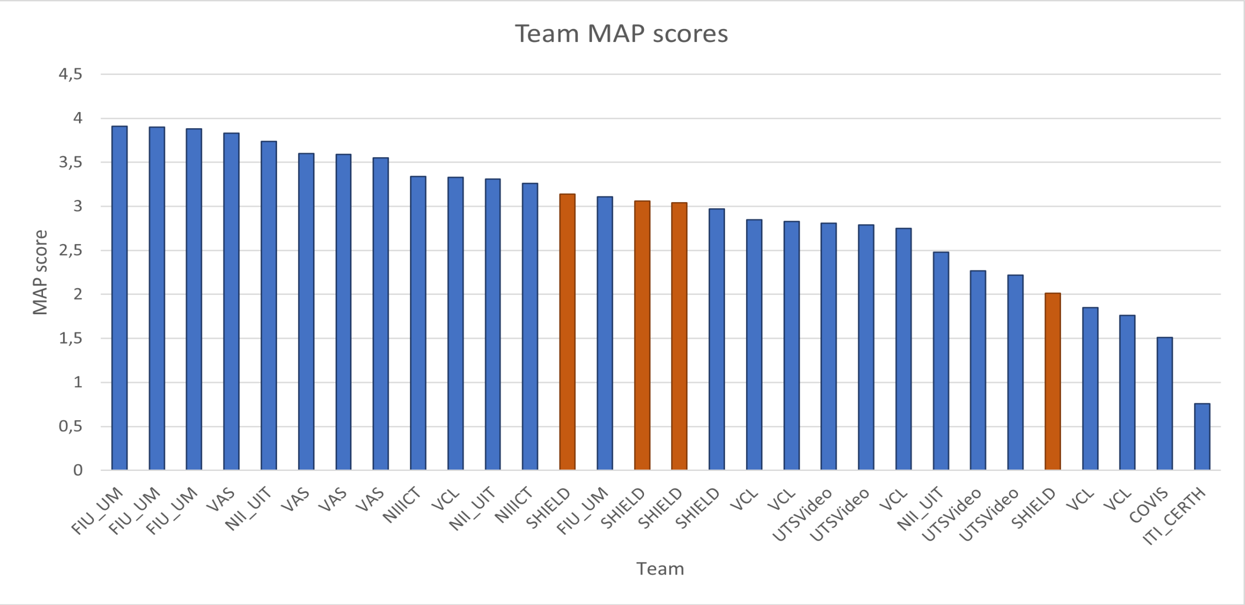


Figure 6: Team scores in term of MAP, our team results for each run are highlighted in red.

| Training Dataset | MAP score |
|---|---|
| LADI + MTurk LADI | 0.314 |
| LADI | 0.306 |
| LADI + OTHER | 0.297 |

Table 1: Best scores obtained during training

## 5 Conclusion

In this paper, we have presented the solutions we have used to address the multilabelling problem on the LADI dataset and also an extension of the dataset that we created using crowdsourcing platforms. After trying various settings we found that the best solution consisted of a model based on five different classifiers, this analysis pipeline is the best we analyzed because it allowed to divide the features by categories and analyze them correctly for multilabelling. And also we found that more samples provided by our extension of the dataset allowed the solutions to perform better. We think that future work on this dataset could have two directions, one first: it is to improve the dataset labels by adding more refined ones, which could cost more resources to create, but at the same time would allow using more sophisticated algorithms. One type of these labels could be segmentation maps. This would allow to specifically learn the areas within the image and to learn which semantic features belong to which area. If this task proves to be too expensive, the segmentation maps could be limited to background elements, such as water and environment, while for the other categories, simple bounding boxes could be used, which in any case would allow for localized learning of the features for that particular class.

## 6 Acknowledgements

This work was partially funded by the European Commission through the FASTER project, grant agreement n.833507, SHELTER project, grant agreement n.821282 and SAFERS project, grant agreement n.869353.

## References

[1] G. Awad, A. A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, A. Delgado, J. Zhang, E. Godard, L. Diduch, J. Liu, A. F. Smeaton, Y. Graham, G. J. F. Jones, W. Kraaij, and G. Quénot, "Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains," in *Proceedings of TRECVID 2020*, NIST, USA, 2020.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[4] K. Crowston, "Amazon mechanical turk: A research tool for organizations and information systems scholars," in *Shaping the Future of ICT Research. Methods and Approaches* (A. Bhattacherjee and B. Fitzgerald, eds.), (Berlin, Heidelberg), pp. 210–221, Springer Berlin Heidelberg, 2012.