Real-time Incidents Detection on Tweeted Images

Alessio Paone
alessio.paone@studenti.polito.it
Politecnico di Torino

Andi Mero
andi.mero@studenti.polito.it
Politecnico di Torino

An Qi
an.qi@studenti.polito.it
Politecnico di Torino

Gianluca Guzzetta
gianluca.guzzetta@studenti.polito.it
Politecnico di Torino

Luca Zilli
luca.zilli@studenti.polito.it
Politecnico di Torino

Sebastian Celis
sebastian.celis@studenti.polito.it
Politecnico di Torino

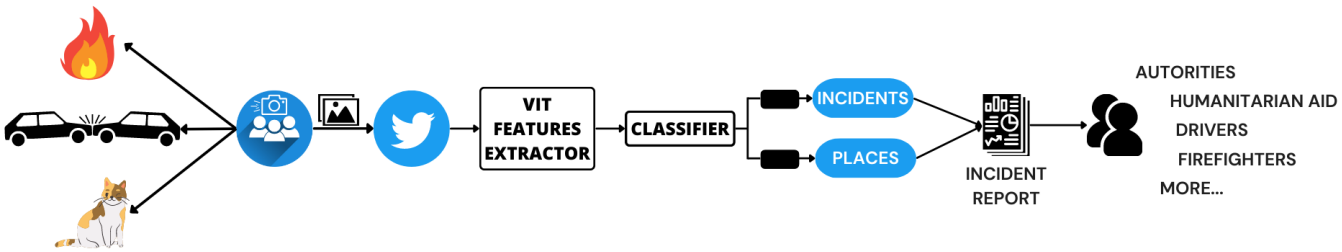

Figure 1: Summary Functional Diagram. Twitter users serve as scouts, posting images that are analyzed via Vision Transformers. The system extracts relevant features and classifies them into incidents and places using a multi-label approach. If an incident is depicted in the image, a comprehensive report is generated and sent to authorities and other stakeholders to improve their ability to perform their duties more safely and effectively. If not, the image is discarded.

ABSTRACT

Due to the effects of temperature and pollution, natural disasters are becoming more common in a growing number of locations around the globe. Third-world regions, for example, are more at risk in case of accidents such as floods, whirls, wildfires and earthquakes. To keep them under control, it is necessary to act quickly and at the same time effectively and safely. We believe that the key to the success of the interventions is the information available to the authorities, who can thus reorganize themselves on the basis of these. This is particularly important in low population density areas, where operations are usually slower due to an insufficient number of emergency units and planning resources. Our aim is to collect this information in real-time and provide a comprehensive report to the authorities through a multimedia streaming pipeline capable of classifying the incidents and the places where occurred based on the images posted on Twitter, i.e. in an Open-Set context. Therefore, we make use of the current SotA (State-of-the-Art) in the computer vision field, represented by the Vision Transformers (ViT) [7], to then compare the performances achieved with those of the paper that inspired our work: http://incidentsdataset.csail.mit.edu [24].

KEYWORDS

Climate change, natural disasters, social media, incident detection, Vision Transformers, Open-Set, real-time image classification.

1 INTRODUCTION

Over the past few years, we have seen a huge increase in the use of social media, and numerous articles have been written regarding the use of Twitter for instant news sharing among other content [9, 14]. With that in mind, social media represent, to date, an infinite source of valuable information (including reports of damages or urgent needs of affected people) that can be exploited for the social good. In particular, where emergency units have few resources in the field, people close to the accident, despite being unable to directly help the authorities, become scouts able to gather information from different points of view on what happened, providing a broader and more comprehensive view of the area. The latter, despite being of great value in emergency situations, would most likely be lost as the authorities are perhaps preoccupied with managing the threat a little further away. For these reasons, the entire development of the project is guided by sustainable development 13 (Climate action) and 15 (Life on land): being able to create a robust and effective solution to better manage these situations could increase awareness of our impact on the climate, reduce it accordingly and preserve those natural habitats that would be destroyed in different types of natural disasters. Furthermore, the content produced by our solution retains its value even after the emergency has ended. For example, it becomes useful material in the training of the authorities themselves, who can use it to carry out simulated interventions on real and specific cases in certain areas, leading to more effective learning. Last but not least, it could serve as a warning for the future to prevent certain tragedies from happening again by defining more precise emergency plans. Our efforts, inspired by the work of the Incidents1M paper [24], have focused on providing an automatic image processor capable of classifying incidents (and the place where they happen) in real time starting from Twitter images in particular. Following their general guidelines, we propose an alternative to the historical CNNs, which uses Vision Transformers [7] as the backbone of our models. The main advantage of this approach is its efficiency and scalability, which allows us to obtain a fine-tuned model with good performance even with limited resources. This is due to the wide availability of pre-trained weights on different

datasets. Even with different augmentation and tuning techniques, our model guarantees results comparable to (or in some cases better than) those obtained with full training on datasets several orders of magnitude larger. Considering our resources, this was in practice a forced choice, which allowed us to "stand on the shoulders of the giants" who provide these resources precisely to pursue future researches. Our survey measure the performance achieved by this models adjusted for the Open-Set scenario on in-the-wild data.

2 RELATED WORK

The current SotA in the computer vision field are Visual Transformers (ViT), first proposed by Dosovitski et al. (2021) [7] and based on the Transformers [22] core architecture, most commonly used in NLP tasks since 2017, when they were proposed by Vaswani. There are many advantages brought by this architecture, which works in a completely different way from that of the more tested and widespread CNNs. Many projects have concentrated on combining the strengths of the two worlds [4, 25], given the higher performance of ViT and the higher inference speed of the classic ResNet for example, leading to hybrid models that appear to be very effective and not far from those of single architectures. In this study, we will use ViT only with pre-trained weights on large-scale datasets using regularization and augmentation techniques. Several studies have attempted to obtain concrete results in the context of emergencies through high-altitude and also satellite images, specializing in specific cases such as fires or floods. Interesting examples in this regard are those for the segmentation of flooded areas [15, 16] or for estimating damage caused by hurricanes [13] or fires, especially in the latter case by predicting the direction in which they will spread based on wind currents [19]. However, a drawback of these works is that they focus specifically on one type of disaster. Furthermore, in several cases they suffer greatly from the occlusion of the area, an immediate consequence of the accidents. Unfortunately this is precisely the time window in which the authorities and humanitarian organizations must act and therefore these sources of information are not suitable for our task. The social media platform has emerged as a fast-sharing collection of last-minute news. Although it is not a traditional source of information, there are few studies using social media images for disaster response [3, 5, 23]. A complete pipeline including duplicate removal and other detailed pre-processing phases implemented on social media images [1] has come up recently. In particular, we are building on the work of Incidents1M [24], which has provided a large-scale dataset of images depicting accidents, including information about the places where they occurred. To complete our work, we examined different method to deploy classification models in the out-of-distribution data, when non-robust approaches suffer the most. We had to work with the assumption that the images might not contain any of our incidents or places classes. The aim is to avoid false-positive predictions: previous works implement approaches such as incorporating negatives directly into training [8] and adjusting a threshold in test phase [12].

3 METHOD

In this section, we present the core architecture of our solution and the implemented precautions used to handle the unbalanced distribution of the classes characterizing the in-the-wild data.

3.1 Architecture

Feature extractor: ViT. Transformers, initially used in the NLP field, consist of two parts: encoder and decoder. However, to use this architecture in the computer vision field, and therefore using images as input rather than text, the decoder part, at least for our task, is not necessary. This was in fact used to reconstruct the output sentences taking into account the context understood by the model. For this reason, in this case, we will only use the *encoder*, which accepts as input a $1D$ sequence of token embeddings. Considering that, in order to use this approach with images, each of them must be reshaped from $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ to a sequence of flattened $2D$ patches $\boldsymbol{x_p} \in \mathbb{R}^{N \times (P^2)}$, with resolution P^2 and C channels. Each of them will subsequently be mapped to D dimensions from a trainable linear layer and paired with a token to retain the positional information, generating so-called patch embeddings. This last step is essential because, unlike CNNs, a downside of ViT architectures is the almost complete lack of inductive bias: in ViT, only MLP layers are local and translationally equivariant, while the self-attention layers are global which means that the model loses the concept of locality and therefore all spatial relations between the patches have to be learned from scratch, another reason why hybrid approaches are still being studied. Finally, the patch embeddings are fed into the encoder, which consists of an alternating series of self-attention layers and MLP blocks. More about the internal architecture reported in detail in [7]. In this study we use this model only as a features extractor (*trunk* model), then classified by two independent heads, one for incidents and one for places.

Model variants. The choice of the size of the ViT model fell on ViT Base, in particular the variant with 16×16 resolution patches (ViT-B-16). Nevertheless, for completeness, in the section Results we also report those obtained from ViT-L-16 (Large), which allowed us to confirm our choice as optimal. In particular, the latter was driven by the constraint arisen from the emergency context, which require, in addition to excellent performance, real-time image classification, forcing us to evaluate trade-offs. In this regards, the resolution of the patches also has an impact on the inference speed and on the performance but, as reported in [20], going up in resolution is preferable only in case you find yourself forced to go down a lot with the size of the model (in other words, ViT-S-32 is better than ViT-Ti-16). For our task, also given the wide variety of scenarios represented in the images, going down beyond the base variant would result in a significant performance loss. The final choice, therefore, fell on ViT-B-16, which is more similar in terms of inference speed to a ResNet50, as it is slightly slower than ViT-B-32 but with much higher performance in some cases like ours in which smaller patches can pick up significant spatial detail.

Classification heads. Once the features are extracted, they pass through two identical but independent classifiers, one for incidents and one for places. These consist of a linear layer which takes as input a feature vector of 1024 dimensions and produces an output vector of 43 or 39 dimensions for the incident and place

classes respectively. The linear layer is designed to perform a linear mapping of the input feature vector to the output class label vector, and it is trained to optimize the classification performance of the model with a customized loss function explained later in 3.2. The linear layer just described, is followed by a non-linear activation function, specifically a sigmoid function, which is used in light of the constraints imposed by multi-label classification. While the use of a softmax activation function would have resulted in the suppression of values associated with less probable labels, ultimately leading to a single possible classification for the image, the utilization of the sigmoid allows for the generation of values within the range $[0, 1]$ for each individual class. These values can be considered as the probability that the image belongs to a particular class. As a result, this approach enables the simultaneous emergence of multiple possible classifications, thereby ensuring a robust but simple multi-classification system. The entire model architecture can be seen in Fig-2.

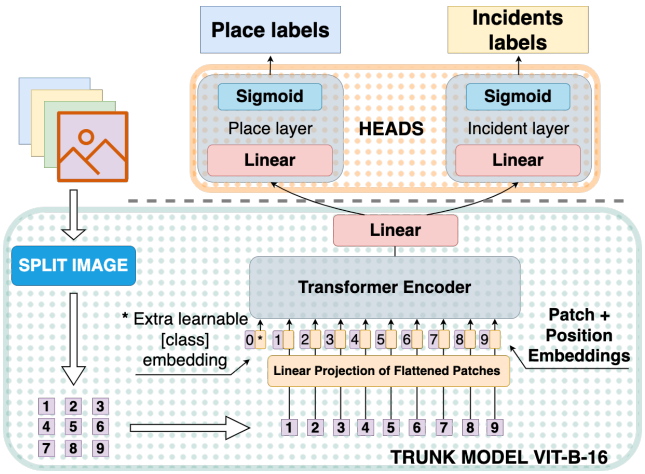

Figure 2: Model architecture overview

3.2 Open-Set context management

Loss. A modified Binary Cross Entropy (BCE) loss, developed in [24], is used to deal with power-law distribution and in-the-wild data. We have to deal, in fact, with very few small images depicting incidents in a very large pool of data and to do it. We need to mitigate false-positive detection, in our specific case both for miss classifications of classes among known ones, and for completely unknown images. In our approach, to train the model, we have chosen to incorporate class negative information, a technique commonly used in object detection rather than image classification and commonly referred to as *contrastive learning*. The purpose of this is to enhance the model's ability to differentiate between normal and emergency situations. Without the integration of class negative images, for example, the model could potentially correlate the presence of vehicles with accidents, even if the vehicles are undamaged. To counteract this, we have included both images of destroyed and burning vehicles as well as images of intact vehicles as class negative examples, in order to accentuate the distinctions between the two situations. Furthermore, by doing this, the model will indirectly tend to cluster the features so as to increase the distance between those of the known classes and, at the same time, form robust boundaries in order to prevent the unknown images from falling into the aforementioned clusters, and thus be misclassified. The formula is as follows (1), and is applied to both classification heads to then sum the output of the two, which will be the final back-propagated value. More detail about this can be found in the Incidents1M paper [24].

$$\mathcal{L}_{in} = \sum_{i=1}^{N} \omega_i (y_i \log(x_i) + (1 - y_i) \log(1 - x_i)) \tag{1}$$

Unknown classification. In the context of an Open-Set scenario, it is imperative to accurately classify images that fall outside the domain of interest and are thus unknown. Several methods have been taken into consideration to address this challenge, including the use of OpenMax [2]. However, due to the complexity of implementation and, in some cases, the total impossibility of doing so due to constraints related to multi-label classification on two different sets, it was determined that a simpler and more effective approach would be to use a threshold on the confidence levels provided by the two classification heads for each class. The final classification will be determined by those classes for which the confidence level exceeds the threshold for both incidents and locations. In the event that no label is recognized, and all sigmoid outputs are lower than the established limit, the image will be considered *unknown*.

4 DATASETS

4.1 Data understanding

Inspired by the work of Incidents1M [24], despite the additional research done, we reconfirmed the use of the same dataset they built. In particular, the availability of this type of emergency images is very limited, especially considering that we do not have the resources to finance the manual labeling of images we have collected. Furthermore, the multi-label feature of this dataset makes it unique and difficult to replicate with the tools at our disposal. The dataset was generated through Google queries and labeled with the Amazon Mechanical Turk platform. It contains over 1.6 million URLs for images with multiple labels for both incidents and places. It was acquired based on the instructions from the authors' public repository[1].

The labels for the categories (incidents/places) contain both positive and negative examples. Figures 3 and 4 depict the distribution of positive and negative examples for the incidents and places, respectively. It is clear how it is a long-tailed distribution, i.e. there is a heavy class imbalance which has to be managed. As for the ratio between positive and negative labels, there are roughly 43% positive and 57% negative labels for incidents and, analogously, around 48% - 52% for the places.

[1] https://github.com/ethanweber/IncidentsDataset

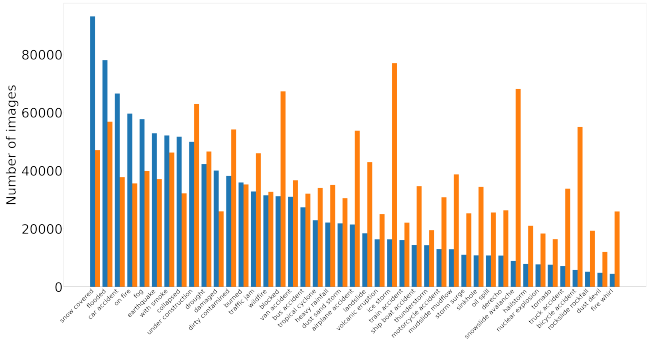

Figure 3: Distribution of incidents classes (blue = class-positive; red = class-negative)

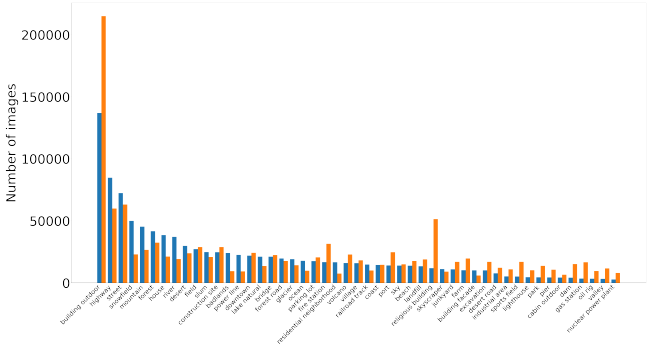

Figure 4: Distribution of places classes (blue = class-positive; red = class-negative)

As previously mentioned, a key feature of the dataset used is the multi-label classification. Unfortunately, Fig-5 shows that the images actually classified in this way are only a minority: most of them contain either no positive labels, i.e. unknown images, or only a single one.

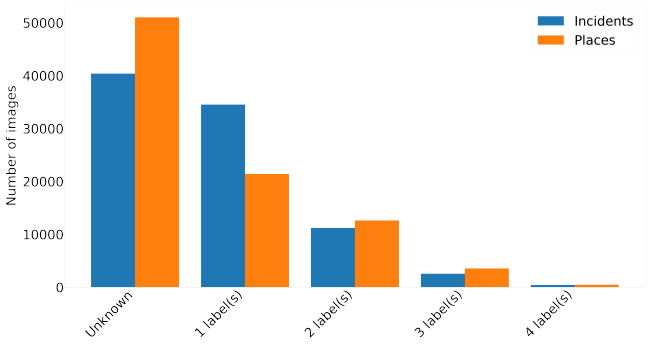

Figure 5: Distribution of positive labels in incidents and places

4.2 Data preprocessing

Once acquired, the dataset looks like a dictionary and the images are only referenced by URLs given the large amount of storage they would otherwise require. So the first step was to download the images still available to have a cleaned, stable and immutable dataset as the links can be removed at any time and the results of each train would be inconsistent. For this reason, downloading them as needed during the different phases was not an option, not to mention the processing time required for each image during training if we had gone this route. With more than 1.6M images in total, the download process was in fact significantly time-consuming and required a large storage capacity, which is why the bwHPC was used (more about that later in 5.1). Fig-6 shows a brief overview of the filtering steps followed: as can be seen, only 1.2M images are actually downloadable from the total and even fewer can be used as input to the model due to corruption during download or invalid format. Finally, we developed a scraper script to filter on-the-fly images based on their classification for both incidents and places for the training phase, as explained later in section 5.2. This eventually resulted in the train-val dataset shrinking down to ~700K images.

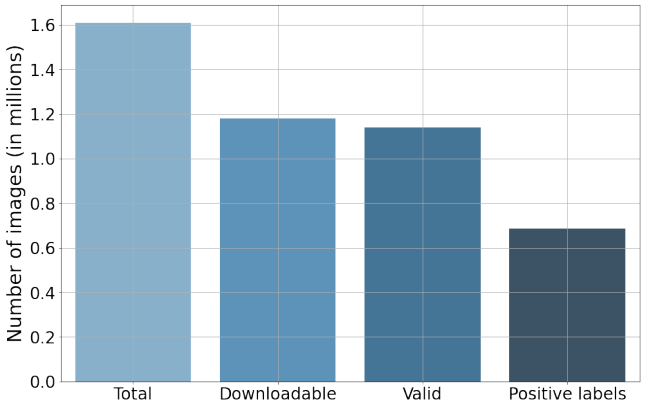

Figure 6: Subsequent filtering steps on Incidents1M (train - val sets)

An analogous procedure was followed also for the test set but leaving in the latter also the images considered unknown, reaching about 60k samples.

4.3 Additional datasets

To confirm our achievements and explore further results especially in the Open-Set context, we considered two additional datasets used only in the test phase: CIFAR-100 [11] and Oxford-IIIT Pets [17]. We employed those two well-known datasets, taking into account that there are no incidents represented, which means we labeled as unknown all the images they contain, simulating the domain of images that the model might encounter on Twitter in a real scenario. Obviously the expected result is that the model classifies all images as unknown which means that we will consider the latter as the positive class to predict.

5 EXPERIMENTS

5.1 Hardware setup

For initial tests and development, Google Colab[2] was used. This tool has proved to be useful for prototyping the system by working as a group in a shared way. However, due to storage and hardware limitations, in order to achieve significant results with almost real-world performance, we used the high-power-clusters from bwHPC [B]. The storage capacity per user on this cluster is 1TB, which is beneficial for dealing with large image datasets like ours. The hardware of the nodes used for training, evaluation and testing are summarized as follows:

- Intel Xeon Gold 6230 (40 total CPU cores)
- 4x NVIDIA Tesla V100 (32 GB of memory per GPU)
- 384 GB of RAM

The hardware information was collected from here[3].

5.2 Training

Dataset. To fairly evaluate the model in the Open-Set context, we trained it on images containing at least one positive label in both incident and place, i.e. excluding the unknown images. In addition to that, given our resources, a sub-sample of the dataset was generated for both training and validation, containing respectively 160K and 40K images. Obviously by doing so, we ensured that we respected the original distribution of the dataset. As the jobs on the bwHPC are scheduled based on the estimated total runtime, this sub-sample size provided the best trade-off in terms of time and performance.

Models. As anticipated in section 3.1, the *trunk* models tested are both ViT Base and Large. In addition to these, as a main comparison we report the results of ResNet50, trained in the same way for a fair comparison. All trunk models have been frozen except for the last layer considering our resources. Regarding the classification heads concerned, they are re-trained from scratch on each trunk model, again for the most fair comparison possible.

Pre-trained weights. We used, for all the models tested in our experiments, weights pre-trained on ImageNet-21k [6] as starting point and subsequently fine-tuned on Incidents1M. The ones for the ViT architectures were downloaded from the Google Research Team GitHub repository[4], while the ResNet50 weights we used are the ones from [18], uploaded on their GitHub repository[5].

Training parameters. Here we report the parameter used to train the models:

- Optimizer: Adam [10], with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0
- Learning rate: 0.0001
- Batch size:
  - ResNet50: 64
  - ViT-B-16: 32
  - ViT-L-16: 8
- Epochs: 20
- Class-labels: positive and negative

[2]http://colab.research.google.com/
[3]https://wiki.bwhpc.de/e/BwUniCluster2.0/Hardware_and_Architecture
[4]https://github.com/google-research/vision_transformer#available-vit-models
[5]https://github.com/Alibaba-MIIL/ImageNet21K/blob/main/MODEL_ZOO.md

5.3 Testing

As previously outlined in the section 4, we conducted testing of the model on a variety of datasets. It is important to recall that validation was performed in a closed set manner, excluding any unknown images. This is because, in theory, images classified as unknown are those that the model has not previously encountered, and therefore they cannot appear during the training phase as anticipated in 5.2. In contrast, during the testing phase, these unknown examples are included in order to provide an understanding of the model behavior in a realistic scenario. Specifically, the Incidents1M test set we made contains 50% unknown images to really benchmark the model.

5.4 Evaluation measures

mAP. It is sensible in case of multi-label classification jointly with the tail distribution of the classes. Precisely, it computes the average precision for each class weighted on the number of true samples of the class itself. We used it in both the validation and the test set of Incidents1M.

F1-Score. It allowed us to explicitly fix a threshold below which the image is considered unknown, effectively reducing the problem to a binary classification between known and unknown images. We used it to evaluate the performance of the model specifically in the Open-Set context on all the datasets at our disposal: CIFAR-100 [11], Oxford-IIIT Pets [17] and Incidents1M [24] (only in the test set as the validation set does not contain unknown images). After some basic tweaking, we set the threshold for the *unknown* classification explained in 3.2 to 0.4 for all our experiments.

6 RESULTS

Initially, we will present and analyze the results obtained from the Incidents1M dataset, which is our primary source of data and the focus of our attention. We will then discuss the testing performed on additional datasets, particularly for the Open-Set context as anticipated. We will start by comparing our results with the original work methodology, and then concentrate on comparing the various ViT architecture implementations to confirm ViT-Base as the preferred choice over ViT-Large.

6.1 Architectures comparison

It is worth reminding that the results of the ResNet50 we reported differ from those presented in the original paper [24] due to our re-training of the model with our data to ensure a fair comparison. As depicted in Fig-7, the two main architectures (ViT and CNN) are closely matched, with our solution performing slightly better without the need for additional fine-tuning. Given the ongoing research surrounding Vision Transformers, much of their potential remains untapped and there is significant room for improvement. Special emphasis is being placed on exploring how to reach performance comparable to that of fine-tuned end-to-end ViT models in downstream tasks with the least amount of extra computing resources. Conversely, ResNet50 has been extensively studied over the years, resulting in well-optimized training parameters and already standard training regularization. Thus, the performance of ViT as an off-the-shelf model is commendable, as it holds its own against an already optimized CNN with no additional research or

modifications which were impossible to perform in this study due to the resources at our disposal.

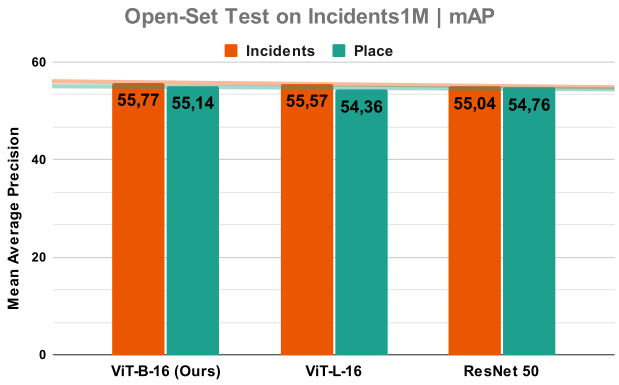

Figure 7: Architectures comparison with unknown images on Incidents1M | mAP metric

The same conclusions can be drawn from the analysis presented in Fig-8, where performance are again similar. The F1-Score is employed, reducing the problem to a binary classification between known and unknown, thereby disregarding the classifier's performance between known classes. Although this view may not be exhaustive, the focus here is on the Open-Set context, specifically on the detection of false positives, reason for which the known class is considered as the positive label. The model shows overall a good behaviour confirming the usefulness of the contrastive loss. The distinction between the incident and place classes is in fact mainly due to the use of a greater number of class-negative examples for the former.

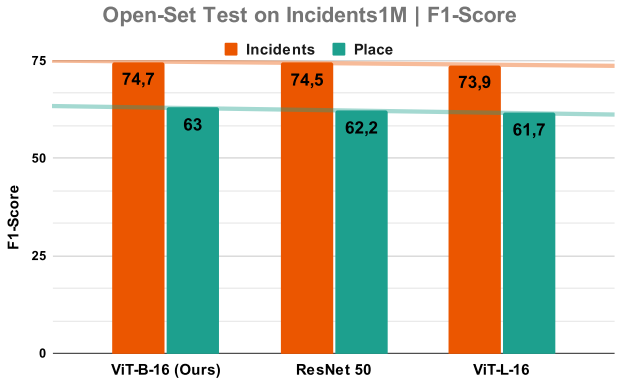

Figure 8: Architectures comparison with unknown images on Incidents1M | F1-Score metric

Lastly, for the sake of completeness, Fig-9 displays also the results from the validation set, which does not contain unknown images. Although our performance falls slightly behind ResNet50, the results are still highly comparable and basically within the standard deviation interval. This highlights even more ViT's capability for better generalization during the testing phase with unknown images, possibly due to its higher number of parameters. However, one can say that ResNet50, with approximately 23M parameters, is a lighter and more manageable model to train and deploy compared to ViT-Base with its 86M parameters. Practitioners must weigh this trade-off when making decisions, but if the objective is to achieve SotA performance, ViT has been established as the preferred model. Moreover, this statement is actually becoming less relevant, as pre-trained ViT models are readily available and can be used with ease for most downstream tasks like we did in this study.
Additionally, the results in Table 1 of the original work [24] show that the end-to-end fine-tuned ResNet50 on the Incidents1M (@25%), which is the most comparable to our scenario, reached 64.01 and 59.68 mAP for incident and place classification, respectively. This, together with the discoveries of recent literature, suggests that, at least with this architecture, performance improvements may be limited. In contrast, compared to traditional CNNs, the advantage of ViT's feature representations lies in their capacity to leverage depth through improved and optimized training procedures as reported in [21]. This provides the potential for significantly surpassing current results. All considered our outcome closely resembles the results from the initial study, with a discrepancy of less than 1.2%, regardless of the use of both known/unknown images and the fine-tuned architecture.

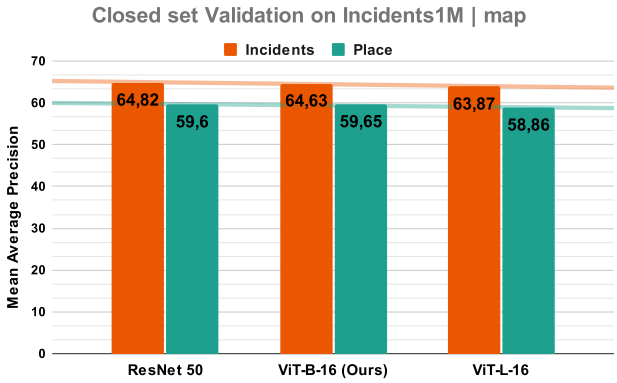

Figure 9: Architectures comparison without unknown images on Incidents1M

6.2 Additional datasets results

Following the method previously introduced in 4.3, we present here the results achieved by our architectures on CIFAR-100 and Oxford-IIIT Pet. The trend in the Fig-10 seems to be the one expected for the latter, which is more uniform and less challenging as it contains only images of cats and dogs. It was surprising to observe a significant decrease in performance for ResNet50, which previously displayed robust results, compared to our ViT-B-16 backbone, with a score of 75.1. However, the situation was different in the case of CIFAR-100, where our model achieved the lowest score of 82. This discrepancy may be attributed to the fact that the VIT Large and ResNet50 models are more prone to confusion compared to the Base variant, generally resulting in lower confidence scores for

class predictions and frequent misclassifications when confidence levels drop below 0.4. Although F1-score takes into account false positive events, it may not be the most appropriate measure in this instance, as the true labels for the additional datasets remain unknown, basically favoring more cautious models. In other words, this particular test may have a slight bias due to the only presence of unknown class, but it also demonstrates the efficiency of the system: instead of making random predictions, the model that is prone to confusion now correctly chooses to classify these instances as unknown.

Moreover it is worth noting that we made an a priori assumption that there are no images of incidents, which could potentially be a false assumption. If this was the case, it would result in a penalization of the model for trying to accurately classify an emergency situation. Hence, care must be taken when interpreting the results of this test, and further in-depth analysis is of course necessary in the future on this topic. To summarize, the results show that ViT demonstrates more consistent performance compared to ResNet50, which has more fluctuating and dataset-dependent results. This reaffirms the reliability of the ViT architecture, even without fine-tuning.

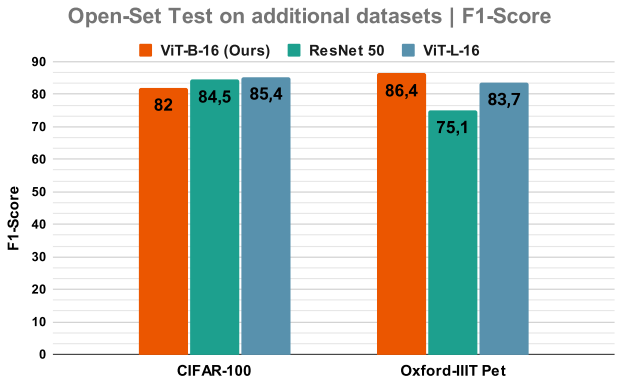

Figure 10: Architectures comparison on the additional datasets

6.3 Inference speed

Even though the performances shown by the ViT models are comparable in our work, we have to consider the trade-off between the overall metrics and the inference speed for the specific emergency context. When dealing with large volumes of images, the Table 1 below illustrates how ViT-B-16 maintains a significantly shorter classification time compared to ViT-Large variant. In this specific field ResNet50 still beats our proposed system by a relatively high margin but this deficiency will be balanced out through better performance once ViT is optimally fine-tuned.

Additionally, smaller variants of ViT could be explored in the future. For instance, as reported in [20], ViT-B-32 outperforms ResNet50 in terms of pure inference speed while sacrificing relatively little performance. However, it is important to note that these tests were conducted on Google Colab, which does not guarantee optimal performance. There are also various methods available to run the model on multiple GPUs in parallel, which would improve performance. In a production environment, obviously the hardware would also permit larger batch sizes, and this analysis aims to compare the relative differences between the architectures, not their actual achievable performance. A more comprehensive analysis considering the trade-off between cost and performance would be necessary to fully understand the topic, but that falls outside the scope of this examination.

Table 1: Average inference speed with Batch size = 4 over 1000 images in total

Architecture	AVG per batch (ms)	STDEV (ms)
ViT-B-16	137.4	4.6
ViT-L-16	501.8	16.4
ResNet50	33.4	8.3

7 CONCLUSION

In this paper, we presented an image classifier based on Vision Transformers technology that is able to classify images in an Open-Set context, making it suitable for real-world applications. Our novel implementation has a meaningful impact on the final system, as it is able to achieve good performance in the natural disaster context, where the images are rarely labeled. The results demonstrated overall comparability with the established ResNet50, with better scores in certain specific cases. Nonetheless, there remains ample room for improvement through future fine-tuning.

As **future work**, we plan to improve the model performance by training it without the computing limitations we had in this study by unfreezing it's layers. Additionally, we would like to investigate methods for providing more detailed information about the images, such as through object detection, metadata analysis and semantic segmentation, as well as exploring techniques for improving the model's robustness to unseen variations of the same class, such as domain adaptation. A really interesting addition would be to integrate other types of data such as **video** and **audio** to provide a more comprehensive understanding of the incident, which could potentially improve the overall performance and usefulness of the system. We are aware that the model has some flaws, and we plan to address them by integrating a module that allows us to see the features the model is focusing on. This could help to counteract false positives by integrating more negative examples of specific cases. Possible extensions to the system include using GANs to infer the damage caused by natural disasters, in order to better assess the impact of these events on artificial constructions, and to raise awareness about our impact on the climate.

Unfortunately, there is still limited research in the field of natural disaster image classification, and we hope that this work can help others and inspire them to develop more effective solutions to aid authorities who risk their lives every day. Overall, this work has demonstrated the potential of using Vision Transformer technology for image classification in an open-set context and specifically in the natural disaster field.

A ABBREVIATIONS AND ACRONYMS

Here the common used acronyms:

- CNN = Convolutional Neural Network
- ViT = Vision transformers
- HPC = High-Performance-Cluster

B RESEARCH METHODS

The work began with forming a group and getting to know one another, including recognizing each other's strengths and weaknesses. To better grasp the topic, we studied related papers and the material provided by our tutor. To keep everything organized, we created a Gantt chart using Asana.

Next, we developed our design by creating Stakeholder and User personas using Miro. This helped us understand the impact our solution would have on their daily lives by exploring their pain points. We also created a service design to map out the user journey. A functional diagram was created to outline how everything would work. This allowed us to refine our Gantt chart and start searching for the best solution.

We focused our research on Incidents1M and preprocessed the datasets, dealing with issues such as broken links, corrupted images, and invalid file formats. Experiments were conducted using bwHPC to compare the results and find the optimal model. We then trained and validated the model and conducted further experiments to fine-tune the results.

ACKNOWLEDGMENTS

We are grateful for the support of the LINKS foundation, who provided us with a prototyping dataset of multi-labeled images to aid in the development of incidents detection.
The state of Baden-Württemberg's bwHPC provided us with the necessary computational resources to conduct the experiments outlined in this paper, and we gratefully acknowledge their support.

REFERENCES

[1] F. Alam, F. Ofli, and M. Imran. 2018. Processing social media images by combining human and machine computing during crises.
[2] Bendale, Abhijit, Boult, and Terrance. 2015. Towards Open Set Deep Networks. arXiv. https://doi.org/10.48550/arxiv.1511.06233
[3] T. Chen, D. Lu, M.-Y. Kan, and P. Cui. 2013. Understanding and classifying image tweets.
[4] Zihang Dai, Hanxiao Liu, Quoc V. Le, Mingxing Tan, Google Research, and Brain Team. 2021. CoAtNet: Marrying Convolution and Attention for All Data Sizes.
[5] S. Daly and J. Thom. 2016. Mining and classifying image posts on social media to analyse fires.
[6] Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2009.5206848
[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv. https://doi.org/10.48550/ARXIV.2010.11929
[8] T. Durand, N. Mehrasa, and G. Mori. 2019. Learning a deep convnet for multi-label classification with partial labels.
[9] Sanja Kapidzic, Christoph Neuberger, Felix Frey, Stefan Stieglitz, and Milad Mirbabaie. 2022. How News Websites Refer to Twitter: A Content Analysis of Twitter Sources in Journalism. (2022). https://doi.org/10.1080/1461670X.2022.2078400 arXiv:https://doi.org/10.1080/1461670X.2022.2078400
[10] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[11] Krizhevsky, Alex, Hinton, Geoffrey, et al. 2009. Learning multiple layers of features from tiny images. (2009).
[12] K. Lee, K. Lee, H. Lee, and J. Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks.
[13] Y. Li, S. Ye, and I. Bartoli. 2018. Semisupervised classification of hurricane damage from postevent aerial imagery using deep learning.
[14] Twitter News. 2022. How many people come to Twitter for news? As it turns out, a LOT. (2022). https://blog.twitter.com/en_us/topics/insights/2022/how-many-people-come-twitter-for-news
[15] K. Nogueira, S. G. Fadel, ´I. C. Dourado, R. d. O. Werneck, J. A. Munoz, O. A. Penatti, R. T. Calumby andL. T. Li, J. A. dos Santos, and R. d. S. Torres. 2019. Inundation modeling in data scarce regions.
[16] K. Nogueira, S. G. Fadel, ´I. C. Dourado, R. d. O. Werneck, J. A. Munoz, O. A. Penatti, R. T. Calumby, L. T. Li, J. A. dos Santos, and R. d. S. Torres. 2018. Exploiting convnet diversity for flooding identification.
[17] Parkhi, Omkar M, Vedaldi, Andrea, Zisserman, Andrew, Jawahar, and CV. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE.
[18] Tal Ridnik, Emanuel Ben Baruch, Asaf Noy, and Lihi Zelnik-Manor. 2021. ImageNet-21K Pretraining for the Masses. *CoRR* abs/2104.10972 (2021). https://arxiv.org/abs/2104.10972
[19] Stefanidou, Alexandra, Gitas, Ioannis Z., Stavrakoudis, Dimitris, Eftychidis, and Georgios. 2019. Midterm Fire Danger Prediction Using Satellite Imagery and Auxiliary Thematic Layers. (2019). https://doi.org/10.3390/rs11232786
[20] Steiner, Andreas, Kolesnikov, Alexander, Zhai, Xiaohua, Wightman, Ross, Uszkoreit, Jakob, Beyer, and Lucas. 2021. How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers. arXiv. https://doi.org/10.48550/ARXIV.2106.10270
[21] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. 2022. Three things everyone should know about Vision Transformers. arXiv. https://doi.org/10.48550/ARXIV.2203.09795
[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017).
[23] E. Weber, N. Marzo, D. P. Papadopoulos, A. Lapedriza A. Biswas, F. Ofli, M. Imran, and A. Torralba. 2020. Detecting natural disasters, damage, and incidents in the wild.
[24] Ethan Weber, Dim P. Papadopoulos, Agata Lapedriza, Ferda Ofli, Muhammad Imran, and Antonio Torralba. 2022. Incidents1M: a large-scale dataset of images with natural disasters, damage, and incidents. arXiv. https://doi.org/10.48550/ARXIV.2201.04236
[25] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time.