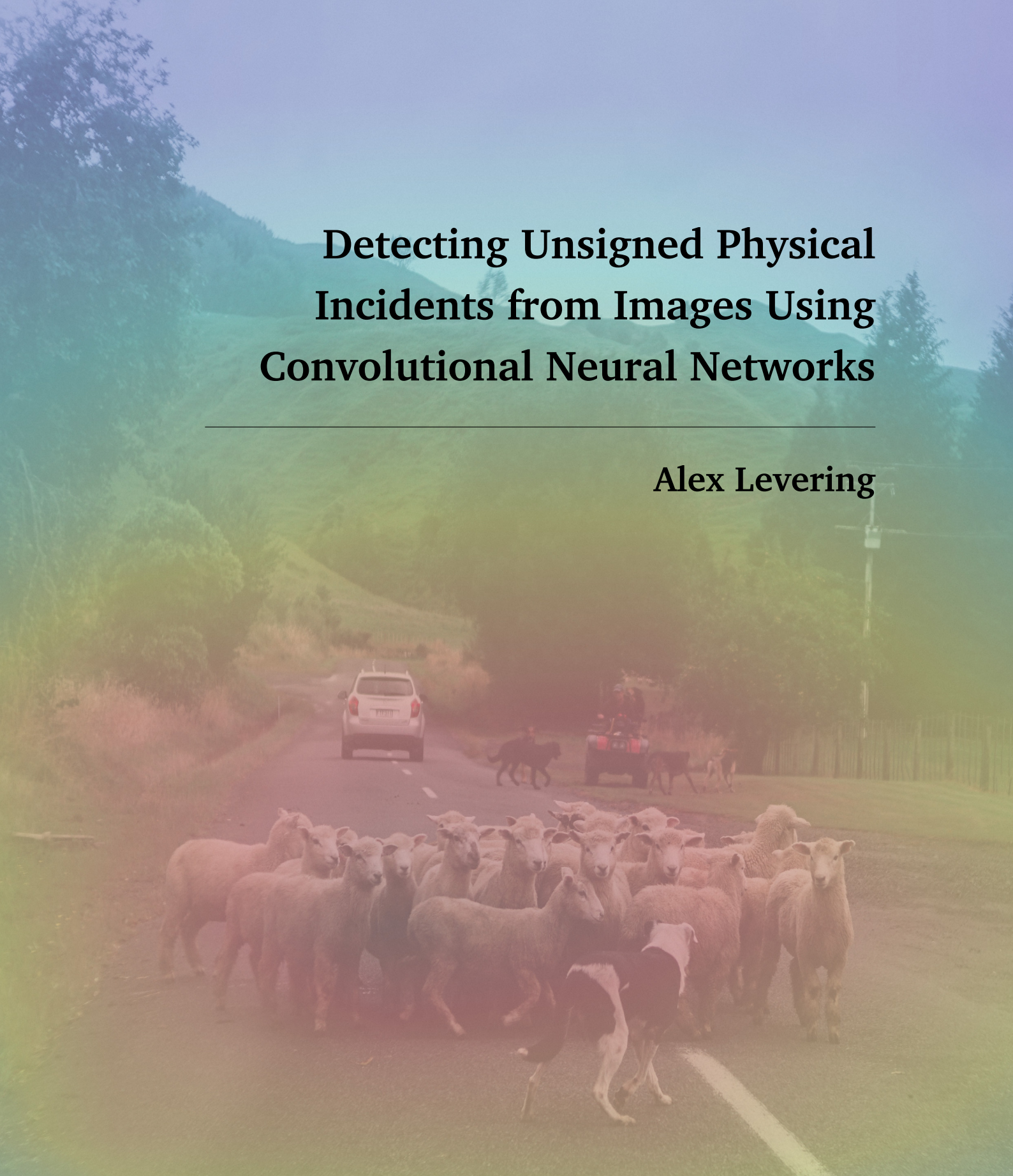
Detecting Unsigned Physical Incidents from Images Using Convolutional Neural Networks

Alex Levering



January 15, 2019
GIRS 2019-01

Environmental Sciences
Laboratory of Geo-information Science and Remote Sensing
Multimodal Remote Sensing Group (MMRS)

Detecting Unsigned Physical Incidents from Images Using Convolutional Neural Networks

Alex Levering

On the cover: A sheep dog herding a flock of sheep on a country road in New Zealand, with class attention for the class *animal on road* calculated using the model developed in this research overlaid on a blue-red colour ramp. Red colours indicate higher attention.

Original photo by Florian Weichelt on Unsplash.

1. Supervisor Devis Tuia
Laboratory of Geo-information Science and Remote Sensing
Wageningen University

2. Supervisor Martin Tomko
Department of Infrastructure Engineering
University of Melbourne

3. Supervisor Kourosh Khoshelham
Department of Infrastructure Engineering
University of Melbourne

January 15, 2019
GIRS 2019-01

Alex Levering

Detecting Unsigned Physical Incidents from Images Using Convolutional Neural Networks

January 15, 2019

Supervisors: Devis Tuia, Martin Tomko, and Kourosh Khoshelham

Wageningen University

Multimodal Remote Sensing Group (MMRS)

Laboratory of Geo-information Science and Remote Sensing

Environmental Sciences

Droevendaalsesteeg 1

6700AA Wageningen, The Netherlands

Acknowledgements

I would like to extend my sincerest gratitude to Dr. Martin Tomko for his excellent guidance during the research. His enthousiasm, patience, and suggestions have been invaluable throughout. I would also like to thank Dr. Devis Tuia for his guidance, trust, and mentorship during the research, especially in the formative and closing stages. My thanks go out to Dr. Kourosh Khoshelham for his proofreading and technical input during the research. I'd also like to thank Benjamin Kellenberg for his advice on the research proposal, Diego Marcos for proofreading the thesis, and my colleagues at the Infrastructure Engineering department of the University of Melbourne for their feedback and suggestions throughout the research. I'd also like to thank Barry Hunter of the Geograph UK Project for providing us access to the Geograph image database.

Declaration

I, Alex Levering, hereby declare that this thesis is the result of my work, and my work alone. All external sources have been referenced and declared where applicable. I also declare that I took reasonable care in ensuring that, to the best of my knowledge, my work does not breach copyright law.

Wageningen, The Netherlands
January 15, 2019

Alex Levering

Abstract

The road network is an active environment which is continuously affected by incidents and disruptions, resulting in delays and economical damage. As car ownership and road transport increases, so too does the pressure on the road network. This results in an increased impact of incidents and disruptions as more individuals and businesses are affected by it. Meanwhile, a recent increase in interest in Autonomous Vehicles (AVs) offers opportunities to lessen the impact of incidents. AVs require up-to-date information on the road network in order to make routing decisions, which promotes the implementation of connected traffic data ecosystems. Such ecosystems allow vehicles to communicate with one another, or with the infrastructure at large to rapidly disseminate information across the grid. In the context of incidents, this means that all connected vehicles can be informed of road closures as soon as they are detected. However, there exists a gap in the literature on incident detection from AVs as a domain. While research considers individual incidents in specific circumstances, no existing research has attempted to classify incidents as a domain or as groupings. This holds true for data on incidents as seen from vehicles as well. As such, a study on incident detection from vehicles that considers the breadth of all possible incidents is needed for the detection of incidents for the purposes of disseminating them between vehicles faster, and thus to lessen the impact of incidents on the road network towards the future. In this thesis we assessed the use of Convolutional Neural Networks (CNNs) to classify unsigned physical (non-placarded, tangible) incidents from street images. We do this by firstly gathering a dataset of images, and secondly by training a CNN to distinguish between images containing unsigned physical incidents and images without such incidents.

Applicable incident classes were determined by a grouping study which made use of a Formal Concept Analysis which resulted in a taxonomy of incidents. In total we then targeted 8 classes: *Vehicle crash*, *Road Collapse*, *Fire*, *Animal on Road*, *Treefall*, *Snowy Road*, *Flooded Road*, and *Landslides*, as well as negatives (images of normal driving conditions). We first collected 7,759 images of incidents by web harvesting from Google, Flickr, and Bing, as well as images supplied by the Geograph UK project. As searching depth-wise (i.e. returning hundreds of images each query)

returned poor results on first experimentation, we decided to perform breadthwise querying by searching for combination pairs between synonyms of various concepts. For instance, query pairs between *street, road* and *landslide, rockslide* yields 4 possible query pairs. 40,063 images have been collected after 118 queries, of which 5,844 images have been included in the final dataset. Additionally, we have submitted queries in various non-English languages to expand the dataset further. We have searched for images using Dutch, Farsi, Mandarin, Croatian, and Slovak by asking colleagues to supply the most effective queries in their own language. In total, we collected 12,630 images over 63 queries, of which 1,641 were included in the final dataset. 5,145 images from the Geograph project were included. Selection of suitable images was done manually by the author to rigorously control the quality of the input images.

After selection of the positive examples, each class is comprised of the following amount of images: [summary of image numbers]. We aggregated a true-negatives dataset of 40,000 images by combining images from Berkeley Deep Drive (20,000), Cityscapes (10,000), and Geograph tagged with *road transportation* (10,000). We also retain 200 negative boundary cases of the class *snow* during the cleaning of Geograph images to help determine whether the model has the correct visual cues. We distribute this dataset into training, validation, and testing splits containing 70/20/10% of all the images respectively. We create a second dataset to test the sensitivity of unsigned physical incident detection to unseen data from different geographical regions by training a second model. We use images supplied by the Geograph project and distribute them into a 72.5/22.5/5% training, validation, and testing split based on the geotags supplied with the images. The training and validation splits contain images from England, Ireland, and Scotland, with the region of Wales being used for the testing split.

Incident detection was performed by training a CNN with the ResNet-34 architecture which performs multiclass-classification over the 8 target classes and the negatives class. The best model achieved a top-1 accuracy of 97.15% and an average unweighted F1-score of 0.8909. We trained and evaluated a second ResNet-34 model for the geographically stratified dataset. The resulting top-1 accuracy for this experiment was 92.9% during testing with an average unweighted F1-score of 0.9169. Assessment of the fully-connected layer of the ResNet-34 model using t-SNE clustering reveals that the model is easily able to tell classes apart. Assessment further revealed that there exists a notable overlap between negative and positive images gathered from the Geograph platform. The results of this thesis indicate that unsigned incidents as a domain can be learned very well. Further research should expand the gathered dataset, consider more incident classes, improve the generated models, perform rigorous bias testing, and experiment with spatial relatedness of features in images (e.g. *animal is on the road* versus *animal is next to the road*).

Contents

List of Figures

List of Tables

List of Abbreviations

AID	Automatic Incident Detection
AV	Autonomous Vehicle
BN	Batch Normalization
CAM	Class Activation Mapping
CNN	Convolutional Neural Network
FCA	Formal Concept Analysis
GPU	Graphics Processing Unit
KL-divergence	Kullbach-Leibler divergence
LIDAR	Light Detection and Ranging
RADAR	Radio Detection and Ranging
RGB	Red-Green-Blue (colours)
SVM	Support Vector Machine
t-SNE	T-distributed Stochastic Neighbor Embedding
UAV	Unmanned Aerial Vehicle

1 Introduction

1.1 Motivation

Traffic is a highly dynamic environment and ephemeral changes to the on-road conditions impact it continuously. Incidents that interrupt the road network can cause economical damage and cripple connectivity, leading to increases in travel time, delays in deliveries, and missed connections to other modes of transportation. Congestion caused by incidents on English trunk roads and motorways is estimated to cost approximately half a billion pounds every year [43], and one in five journeys on these roads result in delays [76]. Furthermore, traffic on England's trunk roads and motorways has grown by over 50% since 1993, and is expected to grow another 31% by 2041 [42]. Combined with the continued trend of car ownership worldwide [25] and a continued increase in road vehicle miles [76], the impact of incidents on the serviceability of the road network may worsen as more vehicles depend on the road network for commutes, leisure, and the transportation of goods.

In recent years there has been a marked increase in the development and interest for Autonomous Vehicles (AVs). The term AV refers to vehicles which possess a certain level of automation, measured on a scale from 0 (no automation) to 5 (full automation in all conditions) [96]. Driver assistance software is becoming increasingly common in mass-manufactured vehicles, with level 1 and 2 AV features (driver assistance, partial driving automation) being readily implemented in many consumer vehicles sold today [72]. Major car manufacturers such as General Motors and Daimler are scaling up their AV production as they undergo the transition from research & development towards mass-manufacturing [72]. Industry research predicts predicts that AV sales will reach 51,000 vehicles sold annually by 2021 and surpass 36 million annual sales by 2040 [48]. Figure 1.1 displays this steep predicted rise in AV sales. In a connected traffic data ecosystem [23], vehicles will be able to further share traffic information about the state of the road network with one, for instance through *Vehicular Ad-hoc Networks* [29], as well as communicating with the infrastructure through Vehicle-to-Infrastructure communication, e.g. as in [69]. Such networks thus make it trivial to distribute information about the road network to any connected car, which greatly improves the ability to disseminate information on incidents as they happen. AVs may make use of connected traffic data ecosystems to retrieve navigation data and updates on the road network status for

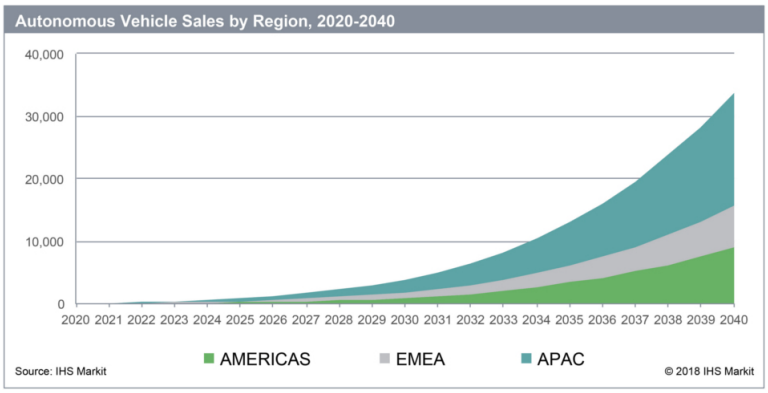


Fig. 1.1: Predicted AV sales in the coming decades per geographical region with the y-axis in millions of sales per year. Image Source: [48].

autonomous navigation, and are thus a major driving force behind the implementation of connected vehicles. The rapid increase in interest from both consumers and producers as well as their implications for our everyday transportation has lead to AVs being regarded as a disruptive technology [14, 28, 93].

While a significant amount of work has been performed on the recognition of 'common' incidents that affect a road network such as sudden pedestrian crossing [32, 30], there is a notable lack of literature on the breadth of incidents that may exist, as well as efforts to create systems to classify such incidents. While less common, these unusual incidents may severely restrain the accessibility of the grid when they occur. For instance, a flooding event or a landslide may close the road, potentially for days. Being able to recognize such incidents may also help to add context to the road network. If a vehicle fire is detected on a particular highway, traffic experts may wish to look at the imagery and determine whether the source of the fire may have other implications such as fumes and bio-hazards.

In response to the increased interest and availability of AVs, both industry and governments alike have started producing infrastructure that accommodates AV functionality. England's highway system has pledged to equip roadways with 5G-connectivity to support Vehicle-to-Vehicle and Vehicle-to-Infrastructure communication, stating that "*The rise of connected and autonomous vehicles is expected to be one of the most significant and potentially disruptive changes in future personal mobility.*" [42, p.19]. Such initiatives allow AVs to communicate trivially with both one another and the network at large. This in turn facilitates recent work by major car manufacturers such as BMW [2] and Mercedes on *High-Definition maps* (HD maps) to complement connected vehicle ecosystems. HD maps allow sensed information from vehicles to be uploaded to the central map which can then be disseminated to vehicles and systems. This is currently done using dedicated sensor vehicles, but in the future the logical next step is to use information derived from road-going

connected vehicles. With a sufficiently large fleet of connected vehicles such initiatives thus allow for live updates and cars that are always aware of the state of the network.

Along with this increase in interest and available technologies there is an ongoing discussion on how safe AVs are. A study concerning the opinion of respondents in English-speaking countries indicates that there exists high concern over the performance of AVs while a majority of respondents also express the desire for them [88]. In March 2018, the first-ever fatal impact of an AV with a pedestrian occurred during nighttime in Tempe, Arizona, causing AV manufacturer Uber to halt all road testing of AVs in that state [66] and renewing the debate on the reliability of the technology used. As is evident, safety is an important topic for AVs.

AVs rely on information derived from sensors installed on the vehicle to interpret the environment around them. Interpretation is performed through a variety of sensors such as LIDAR laser scanners, RADAR systems, optical (red-green-blue) cameras, infrared cameras, and thermal cameras [32], each with their own strengths and shortcomings. For instance, laser scanners can create 3D representations of their environment and provide distance measurements at the cost of requiring more energy and more vulnerable moving components. Cameras recording the red-green-blue (RGB) spectrum on the other hand are considerably cheaper when compared to LIDAR systems and easy to attach to existing vehicles, though they relinquish the 3D representation and the exact measure of distance of laser scanners.

After identifying that connected and autonomous vehicles with sensors equipped and able to recognize a variety of incidents may provide a great benefit to the road network, we aim to recognize incidents from imagery using a well-established information extraction method known as *Convolutional Neural Networks*. With reference to RGB imagery, in recent years there has been a renewed interest in a particular type of image-interpreting methodology. In 2012, seminal work performed in [55] has seen Convolutional Neural Networks (CNNs) become a standard choice for image-interpreting tasks. Their ability to learn what characteristics to look for in an image without manual feature creation has resulted in CNNs widespread adoption for tasks where manually extracting characteristics is too costly. For instance, every year since the re-invigoration of CNNs in 2012, the popular ImageNet [26] classification competition has been won by a CNN variant. This task consist of the classification of 1,000 classes with over 1,2 million images in total with the goal being to achieve the highest possible classification accuracy. Recent developments have seen CNNs become easier to train, faster to process, and more efficient through improvements to their hardware requirements and configuration.

1.2 Research Goals

Based on our research intent stated in the previous section we formulate the following research question:

> *How can we automatically detect unsigned[*] physical incidents from sensors that can be mounted on driving vehicles?*

[*] See p.6 for definition

We formulate our hypothesis for this research question as follows:

> *CNNs can accurately classify unsigned physical incidents in RGB images derived from cameras mounted on driving vehicles*

To test this hypothesis we formulate four research questions:

- **RQ1**: How can incidents be assigned to groupings for the purposes of classification?
- **RQ2**: How do we create an image dataset of unsigned physical incidents?
- **RQ3**: How accurately can convolutional neural networks detect unsigned physical incidents using an image dataset?
- **RQ4**: How stable is the classification of unsigned physical incidents when applying a trained model to unseen geographical regions?

The main purpose of this research is to determine whether incidents are learnable by using CNNs as well as to create a basis for research into incidents affecting the road network. Hence, we do not concern ourselves with exhaustively including all possible incidents or maximizing the classification accuracy as such goals are beyond the scope of this research.

1.3 Thesis Outline

In Chapter 2 we give an overview of the state of the literature on incidents and incident detection, as well as the background on CNNs and their workings. In Chapter 3 we present our choice of methodology and the research set-up that will be considered. In Chapter 4 we give an overview of the most important results of the research, which will be further discussed in Chapter 5. In Chapter 6 we present the conclusions of our research. Lastly, in Chapter 7 we give recommendations for further research.

2 Background

In this chapter we discuss the background material as applicable to this research. Firstly we present an understanding of incidents in existing literature where we define our working definition of unsigned physical incidents. Secondly we discuss concepts relating to image classification and CNNs.

2.1 Incidents

In this section we consider research on incidents and how they relate to roads. We begin by defining unsigned physical incidents before considering past research performed on unsigned physical incident detection.

2.1.1 Definition

In order to delineate incidents as a concept we first discuss the definitions that have been proposed in existing literature. The United States Federal Highway Administration defines an incident as follows: "*An 'incident' is defined as any non-recurring event that causes a reduction of roadway capacity or an abnormal increase in demand.*" [75, p.2]. This definition does not suit the objective of the research as it does not account for a reduction in serviceability, and it attempts to limit incidents to non-recurring events. Incidents under consideration in this research may be recurrent in nature, such as snowfall which occurs seasonally in many regions in the world. Berdica defines an incident as follows: "*An incident is an event, which directly or indirectly can result in considerable reductions or interruptions in the serviceability of a link/route/road network.*" [11, p.118]. This definition is well-suited for this research as it covers for the reduction in serviceability while not delimiting the context in which incidents may occur. As such, we use Berdica's definition as a basis for defining incidents during the research.

In extension to this definition of incidents we delineate the physical nature of incidents. For lack of an existing definition, we consider physical incidents to be incidents that are interpretable by sensors such as cameras, sonar systems, and laser scanners. For instance, a traffic light hack leading can lead to a crash, but the cause of the crash is not easily interpretable using only imagery. By this definition,

non-physical attacks could be malicious attacks made onto a connected network of vehicles, such as packet-dropping attacks (intentionally removing information from the stream), fabrication and alteration of distributed facts, and the creation and dissemination of false information to the network [74]. Consider also the possibility of antagonistic attacks on critical infrastructure [39] such as electricity, which in turn may cripple systems such as traffic lights. Instead, physical incidents concern problems to the road network that can be *sensed* either by one sensor or a combination of sensors. Such incidents can include snow, flooding, stray objects on the road, and vehicle crashes.

Within physical incidents we distinguish between *signed* and *unsigned* incidents. Signed incidents are incidents which are signposted or otherwise marked as a hazard. For instance, roadworks and parades are often signposted with barriers, traffic signs, and high-visibility equipment. On the contrary, unsigned incidents are often ephemeral and unexpected by nature, such as a flash-flood after heavy rainfall. Figure 2.1 gives an example of both types of physical incidents. In this research we only consider unsigned physical incidents. We do not consider signed physical incidents because it shares a broad overlap with highly researched subjects such as street sign recognition, which may result in the task being solved through generalization (e.g. algorithms with the ability to read street signs, and thus hazard markings).

Fig. 2.1: On the left: A signed physical incident in the form of brightly demarcated roadworks [27]. On the right: An unsigned physical incident in the form of a flooding [87].

Enhancing the definition of incidents as proposed by Berdica with the concepts of physicality and signage, the working definition for unsigned physical incidents used in this research will be as follows:

> *An unsigned physical incident is an event without road signs or hazard markers that can be detected by sensors, which directly or indirectly can result in considerable reductions or interruptions in the serviceability of a link/route/road network.*

2.1.2 Unsigned Physical Incident Detection

In this section we consider the various means which have been deployed to detect physical incidents.

Unsigned Physical Incident Detection using Live Sequence Data

There exists a considerable corpus of literature which attempts to classify physical incidents through live sequence data such as live traffic counts. Such research may consider post hoc analysis of sequence data to interpret and gain insight into how traffic incidents affect the road network, such as in [7] where the authors explore visualization techniques to aid decision-making after the incident occurred. More relevant to our research objective is the real-time detection of incidents from a streaming signal such as live traffic counts, such as in [113] where the authors try to predict simulated incidents on freeways and urban arterial roads using expert knowledge employed through a Bayesian network. The end goal of such research is often to predict incidents as they happen. This particular branch of research can be delineated as *Automatic Incident Detection* (AID) [63]. At the turn of the century, AID systems were noted to be performing poorly despite low false-positive rates. Notably, traffic management experts were often detecting incidents at much faster rates than the algorithms could [63]. Live sequence data has also been augmented with other data sources such as video sequence data through data fusion, such as in [112] where the authors investigate the application of multi-source data through Support Vector Machines and Evidence Theory. Beyond improvements in detection algorithms from traffic sequences, recent AID research has expanded to cover improvements in personal connectivity. In [52] the authors consider the use of Bluetooth sensors placed along a highway to detect incidents from trace data. In [85] the authors apply natural language processing to detect incidents from Twitter-sourced data. Research is frequently performed using video sequences, which we consider in the next section.

Unsigned Physical Incident Detection using Sensors

In this section we discuss the detection of unsigned physical incidents using sensor data. We firstly describe incident detection by mounted camera systems before discussing mobile sensor platforms. Here, we discuss the state of the methodology in object detection as relevant to this research before discussing literature on unsigned physical incidents.

Fixed Traffic Cameras

Early work on the detection of unsigned physical incidents predominantly considered the use of fixed cameras such as CCTV cameras mounted on poles to aid traffic experts. In [63] the authors discuss the state-of-the-art of video detection of incidents at the turn of the century, noting that incident detection algorithms on video data are able to reach accuracies between 80 and 90% on live incident detection from traffic cameras. Furthermore, video incident detection performed on traffic cameras could estimate variables such as travel speed, vehicle stalling, and vehicle counts to aid in the decision-making process. However, the authors also note that traffic management experts often vastly outperformed incident detection algorithms on top of having a much quicker detection time. Since then, incident classification systems using traffic cameras have improved notably. For instance, an AID system installed in the Polish city of Gdynia resulted in a 20% improvement in incident response rates after installing a video-based AID system. State-of-the-art commercial classification systems using traffic cameras are reportedly able to reach incident detection accuracies as high as 95% on a variety of incidents, such as a reduction in visibility, vehicle slowing or stalling, shoulder-lane driving, road debris, and congestion [100]. However, research indicates that in spite of these significant improvements in algorithm quality, modern video-based AID system continue to be limited by poor video quality and weather conditions [54].

Object Detection using Sensors on Mobile Platforms

All of the aforementioned research has considered the task of identifying unsigned physical incidents with the intent of monitoring the road network at large. While fundamentally the task of identifying unsigned physical incidents remains unchanged, identification from a road-going ego-vehicle (the vehicle on which the sensors are mounted) has a different end-goal. Instead of identifying hazards with the intent of monitoring a system, research in this domain attempts to identify unsigned physical incidents with the intent of keeping the ego-vehicle safe.

A commonly studied topic in the field of detection from driving platforms is the detection and tracking of pedestrians. While pedestrians are not an incident per se, the vast body of literature on this topic can be used to give insight on the methodologies employed in the domain of object detection and tracking from driving platforms. A comprehensive study of pedestrian identification performed in 2007 records the use of RADAR, LIDAR, mono-and and stereo visible light cameras, and near, thermal, and far-infrared cameras [32]. The majority of the research at the time applied support vector machine classifiers and artificial neural networks. Recent comprehensive research has indicated that a high performance on pedestrian detection using images in the visible spectrum is often achieved using either convolutional neural networks or boosted decision trees (incremented learning of decision trees to overcome misclassifications of previous trees) using manually extracted image

characteristics [114]. The best recorded model only performs a few percent below the human baseline on the tested benchmark dataset. Data fusion techniques to maximize the available sensor information are also widely considered, such as combining camera images with LIDAR [64] or RADAR data [13]. Research into sudden pedestrian crossing is less frequently considered as it requires the inclusion of dynamic factors to separate safe pedestrians from hazardous ones. The authors of [17] attempted to detect sudden crossings from partially occluded regions. They advocate the use of data fusion methods to cover for shortcomings in cameras and LIDAR systems. The authors of [53] have performed near-miss analysis of video sequences with human-level accuracy using semantic segmentation (per-pixel classification) and trajectory prediction using a convolutional neural network. Sudden pedestrian crossing has also been considered during night-time driving [51] where the authors use far-infrared images with manually-selected image characteristics (e.g. textures) and boosted decision trees to overcome shortcomings that hinder cameras recording in the visible spectrum.

Unsigned Physical Incident Detection using Sensors on Mobile Platforms

Research by Zhou et al [118] considers the state of animal detection systems for road-going vehicles until 2014. They record five research efforts that are applicable to intelligent vehicles which intend to classify various species of animals. However, most recorded research efforts concern general animal classifiers which may be applied in intelligent vehicles rather than specific classifiers for animals on the road. Directly relevant to unsigned physical incidents is the research effort by Saleh et al [86] where researchers have detected kangaroos on the road from images through semantic segmentation with the express purpose of collision prevention.

Other types of unsigned physical incidents are scarcely considered in existing literature. Research by Chen et al [20] considers the use of LIDAR sensors for the detection of road obstacles by defining the driving surface, then detecting outliers on this driving plane. In [59], the authors approach this same task by using images from RGB cameras to solve for the perceivable edges of the driving surface in order to find the ground plane boundaries of potential obstacles on the driving surface. However, none of the research attempts to define the object that is obstructing the driving path, and neither method distinguishes 'regular' obstacles such as cars from unusual obstacles such as debris. To our best knowledge, the only research in this domain that defines the type of hazard to occur on the driving surface is [91] in which the authors attempt to detect shallow holes and water hazards on the driving surface using the (lack of) returns of a given horizontal beam of a LIDAR scanner.

In conclusion, while there exists a considerable body of literature on the detection of unsigned physical incidents, such literature often does not pertain to driving

platforms, and less frequently attempts to define the type of incident, let alone consider multiple incidents at once. To our best knowledge, no literature has been published that considers incidents beyond a vaguely-delineated concept, such as animals or obstacles. Therefore, a typology of incidents has to be created in order to adequately consider the breadth of incidents that may occur on a road network. Furthermore, the lack of research in this domain results in a lack of available data, and therefore a data gathering effort has to be undertaken so that incidents the incidents under consideration may be recognized from sensor data.

2.2 Image Datasets

Concluding from the previous sections there thus exists a lack of datasets on unsigned physical incidents, which prompts the necessity to generate a dataset during the research. In this section we describe notable existing research on generating large-scale image datasets for the purposes of classification in order to understand best practices for dataset creation. In the interest of relevance we only consider web-gathered image datasets in this section.

Large-scale datasets using web-gathered images for classification have been widely considered in computer vision studies. In 2009, the National University of Singapore released the NUS-WIDE dataset, which consists of 269,648 images and their associated tags from Flickr. This research attempted to build a dataset through the gathering of Flickr images which they test against ground truth generated for 81 high-level concepts (e.g. *building*, *car*, and *waterfall*) from volunteers from several high schools and university colleagues. They found a significant amount of noise (incorrect images) in some concepts but very low noise in others. Overall, the signal-to-noise ratio was found to be around 50%. With a total of 5,018 unique tags it was the largest dataset at the time. The authors do not discuss storage and dissemination practices for the dataset.

Later in this same year Deng et al [26] released the ImageNet dataset consisting of 3.2 million images across 5,247 synsets, which are sets of synonyms and depth-wise related concepts. The implementation of synsets is based on Princeton University's WordNet [68]. The ImageNet database employs a tree structure which groups synsets at various semantic levels. For instance, the *husky* class is a subset of the *working dog* grouping, which is in turn a subset of *dogs*. It is furthermore notable for being a dataset with a high signal-to-noise ratio, in which it differs from other datasets at the time. An average of 99,7% precision is achieved for each different synset. Data was collected by querying "*various search engines*" [26][p.4] with queries consisting of the semantic concept and appending their parent synset name.

Furthermore, images are translated to Chinese, Spanish, Dutch, and Italian to form multilingual queries. Images are subsequently cleaned using Amazon's Mechanical Turk service [6], where users are paid to perform tasks which require human judgment, such as deciding whether an image fits a set of criteria. For a given batch, each image was presented to the worker with its synset term and Wikipedia definition, and the user was asked whether an image is acceptable for the term or not. They also successfully circumvent low accuracy on difficult semantic concepts by showing the same image to multiple users so that a consensus can be reached. Images are disseminated through *www.image-net.org* and are supplied as image URLs so that users can download their own data. Images are also supplied as a direct download, in which case images may only be used for non-commercial and research purposes as the authors of ImageNet do not possess the copyright of the gathered image material.

Recent developments on the collection of large image datasets has seen image datasets generated with more images for a smaller subset of classes. To create the LSUN dataset [110] the authors attempted to expand on a small subset of synsets with a similar methodology as ImageNet. The authors use manually selected adjectives to perform synonym searching using the Google Images API. For the 20 selected subsets they collect millions of images through a similar mechanical turk set-up as ImageNet. Their methodology differs from ImageNet's methodology in that they supply ground truth images along with their usual batches. Batches of images were only accepted if the average accuracy of the batch exceeded 85%. Overall, the ImageNet dataset is more is more accurate, but the LSUN dataset contains a greater number of images in total, gathering 1 million images for each synset for a total of 20 million images. The dataset is supplied through a *Lightning Memory-Mapped Database*, which is a fast key-value storage database. It only stores the URLs to the collected images.

2.3 Image Classification

In this chapter we discuss concepts relevant for scene classification before considering convolutional neural networks (CNNs) as a methodology to apply for this task.

2.3.1 Supervised Scene Classification

From the standpoint of engineering, the primary purpose of computer vision is to create computational models of tasks performed by the human visual system [83]. The classification of images into distinct groups is one such task. Classification as a general task can be understood as "... *the problem of identifying which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations.*" [3, p. 39]. In this research we perform a task known as *Scene Classification*, specifically single-label classification where the constraint is that for every image there is only one label associated with it. This task has been formulated by Boutell et al [16] as follows:

> "*Let X be the domain of examples to be classified, Y be the set of labels, and H be the set of classifiers for $X \rightarrow Y$. The goal is to find the classifier $h \in H$, maximizing the probability of $h(x) = y$, where $y \in Y$ is the ground truth label of x.*" **Boutell et al. [16, p.1]**

Single-label classification can thus be approached as a supervised learning problem, as labeled data can be used to train a classifier with the explicit task of maximizing the probability $h(x) = y$. Hastie et al [40] describe supervised learning as follows:

> "*Supervised learning attempts to learn* (function) *f by example through a teacher. One observes the system under study, both the inputs and the outputs, and assembles a training set of observations $T = (x_i, y_i), i = 1, ..., N$. The observed input values to the system x_i are also fed into an artificial system, known as a learning algorithm (usually a computer program), which also produces outputs $\hat{f}$ in response to the inputs. The learning algorithm has the property that it can modify its input/output relationship $\hat{f}$ in response to differences $y_i - \hat{f}(x_i)$ between the original and generated outputs.*" **Hastie et al. [40, p. 29]**

Supervised classification requires the use of a training dataset. This is the set which the classifier gets to use to improve its performance by tuning its parameters to maximize the amount of correctly predicted examples. Secondly, a validation set is used against which the classifier is measured without learning from its mistakes.

This *validation* dataset is used to continuously check whether the classifier is able to perform well on new, unseen data. Once a classifier is trained that performs well on the validation dataset it is then tested against the *test* dataset. This dataset is evaluated only once in the development of the model, and serves as the final accuracy check on new, unseen data.

Following the aforementioned principles, datasets have to be divided into three *splits*; *training*, *validation*, and *testing*. Typically, each set contains progressively fewer examples. Every image included in the training dataset is another image that the algorithm learns from. However, taking away images from the validation dataset would mean a weaker insight onto how well the model performs on unseen data. If there are too few images in the test dataset then the final reported results of the tested model may be uncertain as a conclusion is then drawn on too few examples. Most published research allocates between 70 and 80% of the data for training and 5-10% of the data for testing, with the remainder being allocated to the validation split.

In order to effectively classify images to labels, it is firstly necessary to extract *features* from these images. A digital image with red-green-blue colour bands is in essence a matrix with three dimensions and hundreds of rows and columns containing information on a wide variety of topics such as hue, contrast, geometric features, and brightness. These features are muddled and compressed to the same three dimensions. In order to classify images based on information from these images it is worthwhile to extract information from these images first. It is therefore necessary to perform *Feature Extraction* so that the chosen characteristics of the image (e.g. colours, textures, and edges) adequately describe the phenomenon to be classified. Many feature extraction techniques exist, such as first & second-order edge detection, image motion descriptions, shape matching, texture extraction, and statistical features [84].

Features extracted from images can then be used to predict a label by using a classification algorithm, also referred to as a *classifier*. The *Softmax* classifier is one such classification algorithm. It is given as follows:

$$p_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{2.1}$$

where p_j denotes the probability of a class j given an input vector z, which may for instance be the feature maps computed using feature extraction. In doing so, images can be assigned to a particular class based on the features extracted from the image matrix. This function can be optimized by calculating the *cross entropy* loss

function during the training phase, where the classifier can learn from examples. Loss is the difference between the predicted label and the ground-truth label [4]. The algorithm for cross entropy loss is given as follows:

$$L_i = -f_{y_i} + \log \sum_j e^{p_j} \tag{2.2}$$

Where $p_y i$ is the predicted probability for the true class label. The Softmax classifier combined with cross entropy loss attempts to minimize the estimated class probabilities and the actual class distribution. If a classifier does not perfectly predict the label of an image (a confidence of 1 on the correct label) it will incur loss. The Softmax classifier incurs more loss the lower its confidence for the correct label is. By optimizing the classifier to have a minimal amount of loss, the Softmax classifier is thus trained to want all of its confidence on the correct class every time.

In the next section we discuss Artificial Neural Networks, which is a methodology that combines both feature extraction and classification.

2.3.2 Artificial Neural Networks

Feature extraction is a process that may take up considerable time and requires expert knowledge on the topic. With so much information compressed into three dimensions, which are the right features to use to train a classifier on? Artificial Neural Networks (ANNs) offer a solution to extract information from the image matrix without manual intervention. The elementary form of ANNs contains only an input layer with an activation function and subsequently a classifier. One early implementation of this concept is known as Rosenblatt's Perceptron [81] which is given in Figure 2.2.

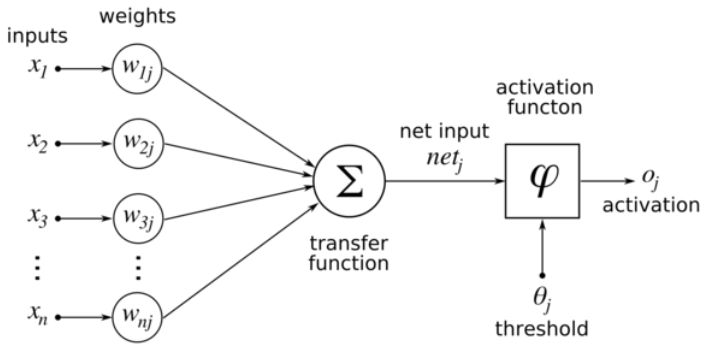


Fig. 2.2: Rosenblatt's Perceptron, where X_j represents numeric inputs, W_{nj} represents weights applied to the inputs, θ_j represents the threshold which determines activations, and o_j is the resulting vector of computed values. Image Source: [70]

Once the weighted value satisfies the threshold of the activation function, the neuron of the input layer activates with a positive value and is thus included in the classifier. For instance, we might have a binary activation function; If the value is above 0, we activate the neuron (assign it a value of 1). Otherwise, its value is zero. Such functions lets neural networks 'choose' which parts of the input data to use to solve a particular problem. In doing so, each neuron of a layer can store separate bits of information as their weights can be tuned individually and thus learn different representations. After the activation function computes which features are present in the input data, the activated features can be used in a classifier such as a Softmax classifier. To emphasize this point, neural network layers make up a trainable feature extractor to use in a classifier, which is attached as a separate layer at the end of the network. Using the loss derived from the classifier, the network can then be tuned to provide better features for the classifier to use.

However, the above algorithm was found to be insufficient to learn more complex representations due to its linear nature, and soon after researchers began experimenting with Perceptrons with multiple layers [50]. Yet, in order to start learning sequential layers effectively a means of efficient automatic tuning of the weights for each layer had to be invented. This came in the form of backpropagation, the basis for which was first conceptualized by Werbos [108] and later popularized by Rumelhart et al [82]. The resulting gradient signal can then be used to tune the weights of any given function by updating it relative to its derived gradient. Simply put, backpropagation is a means of deriving which neurons caused the computed loss value, and to update neurons with regards to their success or failure accordingly.

The gradient derived from backpropagation can then be used to optimize a network. The simplest form of optimization for neural networks is gradient descent [19], which is an iterative optimization algorithm which can be used to find the minimum of a function. In the context of a neural network, this is a function F can be parameterized by the model's parameters $w \in M$, where the desired output of the function is to correctly classify every training sample. On each iteration, gradient descent computes the gradient of the function as $\frac{\partial}{\partial w} F(w) = \nabla_w L$. The gradient computed can then be updated as $w = w - \alpha \nabla_w L$, where α is a hyperparameter known as the learning rate. This same process is performed for the biases b of each layer. The minimum here represents the global minimum of the loss function, which can be expressed as a high-dimensional polynomial governed by the network weights [21]. While this local minimum is not necessarily the best solution (which is referred to as the global minimum), modern deep neural networks are empirically verified to be equivalent to other local minima, and thus yield similar performance on unseen data [21]. Intuitively, gradient descent can be understood as a ball rolling down a hill. The fastest way to reach the bottom, and thus the

global minimum, is to find the steepest slope to roll down on. Figure 2.3 expresses this visually. At every iteration the function computes the gradient for the current timestep x, then updates the weights so that at the next iteration it will roll in the steepest downward direction, thus minimizing the function. Iteratively doing so will lead to a network that has converged onto its lowest local minimum on the loss surface. More sophisticated optimization algorithms exist which aim to solve problems with gradient descent, but for the sake of brevity we do not discuss them in this background chapter. For more information on optimization theory we refer the reader to Weise [107].

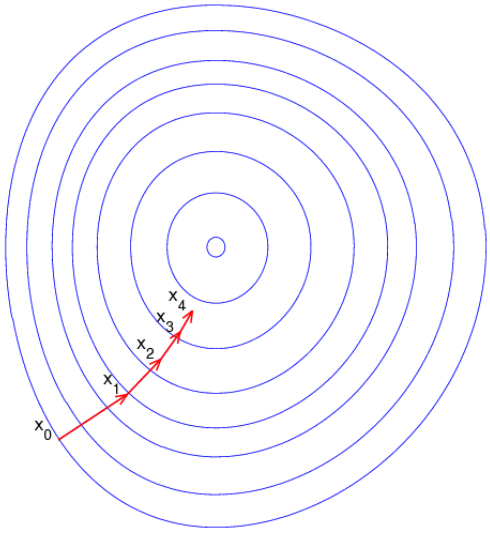


Fig. 2.3: Gradient descent through subsequent updates, where each x represents a timestep. Image Source: [5]

Notice that the process of gradient descent can only be performed for a single function, and thus a single layer of a network. To extend this concept to deeper layers within the network, chain rule derivation was introduced to derive the gradient at each layer of networks that feature multiple layers. Chain rule derivation is a means of deriving for a variable within a composite function. It is given as follows:

$$F'(x) = f'(g(x)) \;\; g'(x) \tag{2.3}$$

Where F and G are functions, and x is the variable to solve for. Within neural networks, each layer can be interpreted as a function that transforms a given vector of inputs to a new vector of outputs by applying a transformation to each element [80]. Assuming that each of these layers are derivable, we can thus derive the gradient for a particular neuron by first deriving the layers between it and the loss

function. This process is repeated for each neuron in the network all the way up to the input layer.

Updates performed using gradient descent take into account all data points at once. With large datasets, this causes problems with virtual memory. To remediate this problem, Stochastic Gradient Descent (SGD) was conceptualized. SGD relaxes the need to use the entire dataset during training by instead performing gradient descent using a randomly chosen data point from the entire dataset. This makes it possible to iterate through the examples and perform gradient descent at each data point [15].

With SGD and backpropagation using chain rule derivation it became possible to use layers in a network. Networks using multiple hidden layers are known as Multilayer Perceptrons. On top of the input layer with its activation function, such networks contain one or more fully-connected hidden layer which uses the weighted sum of all the outcomes of the previous layer combined with a non-linear activation function, as shown in Figure 2.4. This non-linearity lets networks compute more complex (non-linear) features [80].

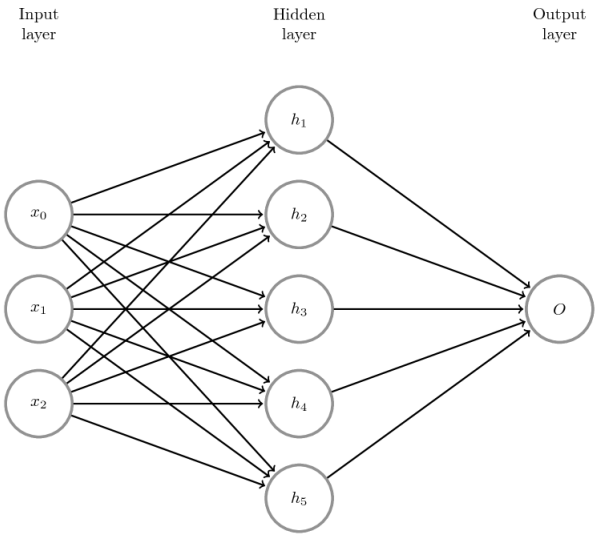


Fig. 2.4: Multilayer Perceptron with one hidden layer. Notice how each neuron in the hidden layer uses the combination of all of the input neurons. Image Source: [65]

The next section discusses the variant of neural networks which was invented to deal with the problem of extracting features from image matrices.

2.4 Image Classification using CNNs

Convolutional Neural Networks were conceptualized as a variant to traditional neural networks to handle information extraction tasks from images. Traditional neural networks were not widely considered for this purpose due to their globally-connected structure. Layers with global connectivity analyze the entire pixel space for different features at each location, which is wasteful as neighboring pixels in images are typically correlated and patterns that appear in images often appear in various places across the image space. Furthermore, the amount of parameters required per layer is unmanageable for deeper networks. For instance, a layer with 3 color bands and 200×200 pixels uses 120,000 weights for a single feature. Clearly, this results in poor scaling if many such features have to be extracted. Layers with complete connectivity between layers are known as *fully-connected layers*.

The pursuit of conceptualizing networks with more manageable parameter counts lead to the invention of Convolutional Neural Networks (CNNs). Conceptualized by Fukushima et al in 1980 [31] and popularized by LeCun et al [58] eighteen years later, CNNs use locally-connected layers known as *convolutional layers*, from which the network architecture derives its name. Convolutional layers slide *filters* of $P \times P$ pixels spaced D pixels apart across local portions of the image which are multiplied with the input image at each location. This process is displayed in figure 2.5 This produces matrices that contain values indicative of features that may be present in images, such as an edge, contrast, or colour adjacency. To further reduce the parameter load, each filter is computed globally across the image space. That is, one filter checks every section of the image for the same feature. The number of filters can be chosen per layer, and as such a convolutional layer can compute many features such as an edge detector and a color adjacency filter all within the same layer. On top of reducing the parameter count per feature computed, their theoretical accuracy is only slightly worse than fully-connected networks [55]. Convolutional layers are followed up with an activation function in the same way that fully-connected layers are. A commonly-used non-linear activation function is the ReLU (Reactified Linear Unit) non-linearity, which sets any negative input value to zero.

Many CNN architectures use pooling layers at various places in the model. Just like convolutional layers, pooling layers consist of a sliding window of $P \times P$ pixels spaced D pixels apart. Pooling layers summarize information by keeping a statistic about the window being considered, such as the maximum value [55]. At each filter position the pooling layer extracts a value according to a criterion (e.g. maximum or average value), then assigns this value to a single output pixel. Figure 2.6 demonstrates this for a 2×2 pooling layer with max-pooling. Effectively, 2×2 max-pooling halves the resolution of all feature maps, and in doing so it significantly reduces the

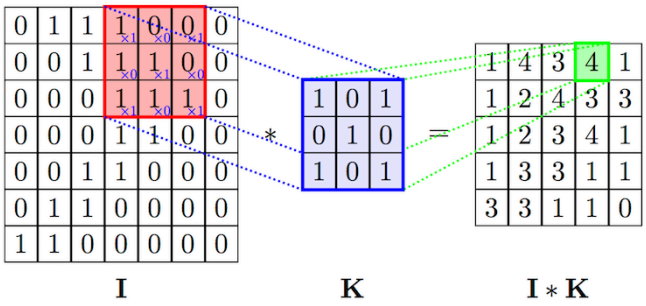


Fig. 2.5: An example of a convolutional filter, where I is the input image and K is the learned filter. Notice how the resulting multiplication makes up a new, smaller image. Image source: [104].

computational load for subsequent convolutional layers, as well as filtering noisy activations and allowing wider extents to be viewed by small filters.

Fully-connected layers, convolutional layers, pooling layers, and ReLU activations form the basis for modern CNNs. As an example on how these components look operationally, Figure 2.7 displays the LeNet-5 architecture [58] which is a CNN architecture that uses all of the aforementioned components. Layers denoted with C are convolutional layers. For instance, the first layer learns 6 filters, resulting in 6 activation maps. Layers denoted with S perform subsampling, sizing down the images by averaging all values within their 2×2 sliding window. Lastly, activation maps are passed through two fully-connected layers before being used as output in the classifier.

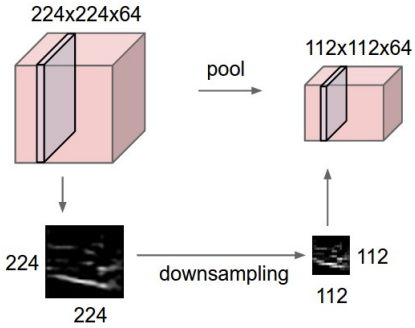


Fig. 2.6: An example of a max-pooling operation, where all 64 feature maps in the system is subjected to a 2x2-pixel sliding filter, taking the maximum pixel at each location in a 2x2 grid in the operation. Image source: [60].

Modern published CNN architectures see both SVMs and Softmax classifiers in use. The classification accuracy offered by the choice of classifier does not seem to vary greatly, though it has been suggested that SVMs may offer a slight improvement in accuracy over Softmax classifiers [99]. Nevertheless, the majority of benchmark CNN architectures use the Softmax classifier as the output classifier of choice, such as in [98, 41, 46].

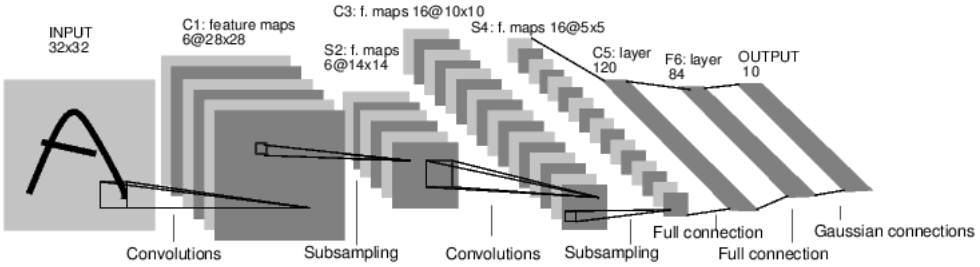


Fig. 2.7: The LeNet-5 architecture which uses two series of convolutional layers followed by a Tanh activation function and an averaging pooling layer. The Gaussian connections refer to the output classes for classification. Image source: [58]

CNNs are often limited by their data availability. Because CNNs learn from their input data, a large amount of input data is required to learn filters that generalize well to unseen examples. For instance, a model that has only been trained on daytime images will likely perform poorly on nighttime imagery, as the filters may not trigger due complications such as missing color information or a lack of clearly visible edges. As a result, image datasets for CNNs are desired to be in the range of thousands to millions of examples in order to ensure that the model is robust and generally applicable. One such example is the ImageNet dataset, which contains 1.2 million images covering 1,000 target classes [26]. Nevertheless, even such large amounts of data may prove insufficient for training deep CNNs with millions of learnable parameters. One technique to increase the potential of the network to generalize beyond the given training dataset is to apply *augmentations* to the input data. Augmentations refer to deliberate deformation and transformation of the input image matrix. Examples of such augmentations are geometrical operations such as rotations and shearing, as well as operations that adjust the image appearance such as greyscale transformations, contrast, and adjusting hue. Controlled experiments have shown that classification accuracy improves slightly after applying such transformations [77, 22].

To summarize, convolutional neural networks are a methodology that combines both feature extraction and classification by using filters to learn feature representations, using these features in a classifier, and subsequently they use the loss of the classifier to provide better features to use in classification. This powerful combination saw highly accurate CNNs become the defacto standard for many classification-related computer vision benchmarks such as classifying the popular ImageNet dataset [10]. Beyond hypothetical classification scenarios, CNNs have been widely applied in domain research cases. In the domain of incident detection they are often used for computer vision tasks, such as near-miss detection [53], animal detection [86], and road boundary detection [59].

2.4.1 Influential Network Architectures

In the past couple of years models have progressed in their layout and design practices. Best practices evolved, and so too did the efficiency of CNNs. Several notable network architectures have been developed that furthered insights into their workings.

After LeCun et al invented the LeNet-5 architecture [58] in 1998 there was a gap in the development of CNNs, which lasted until 2012 when Krizhevsky et al [55] conceptualized the AlexNet architecture. Its base architecture is similar to LeNet-5 though it used more layers and more modern insights in its design. It featured 5 convolutional layers which used 11x11, 5x5, and three consecutive 3x3 wide filters with an input image size of 227x227 pixels. It also substituted average pooling for maximum pooling. Aside from reinvigorating interest in CNNs, a second notable contribution was to train the model on a Graphics Processing Unit (*GPU*). GPUs had previously been identified to speed up tasks requiring linear algebra [57], which neural networks make use of in the form of matrix multiplications (convolutional filters, pooling layers). The speed-up gained through GPUs made it feasible to train deeper CNNs than had previously been the case. The AlexNet architecture also made use of ReLU non-linearities, which peg negative values to zero, leaving any positive signal untouched. Krizhevsky et al explored the effect of deeper sequential convolutional layers and advocated their importance, noting a drop in accuracy if any of the five convolutional layers in the model were removed.

Another noteworthy architecture is GoogLeNet which was invented in 2014 [98]. GoogLeNet explored the concept of parallel convolutional layers with different filter sizes in what was dubbed an *inception module*. They also attempted to address the vanishing gradient problem, where consecutive layers in deep neural networks receive decreasingly little gradient [45], by using not one, but three Softmax classifiers at various locations in the model so that earlier layers could also receive signal. In doing so, they could train deeper networks than had previously been the case. Figure 2.8 displays the architecture with one of the inception modules highlighted.

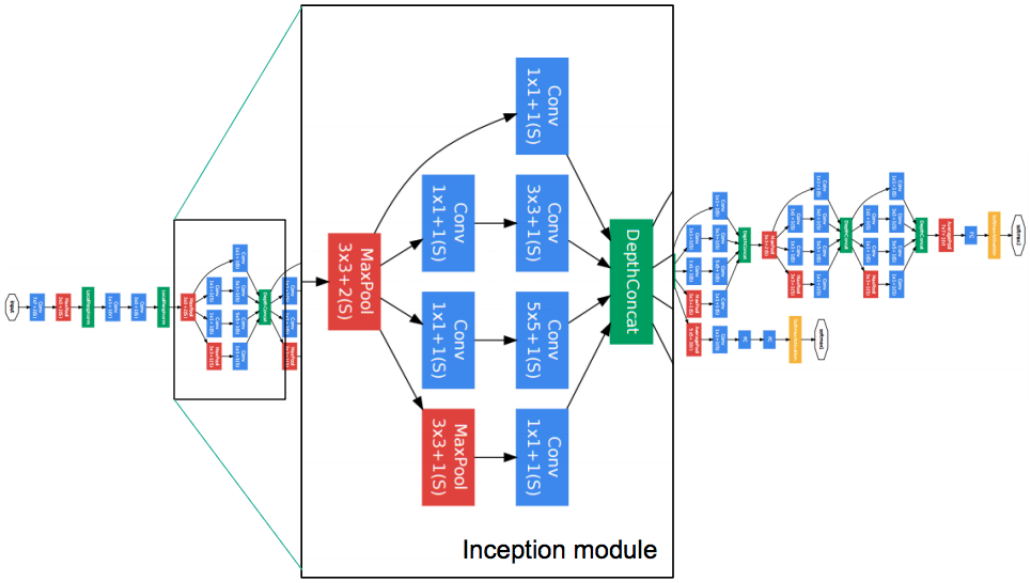


Fig. 2.8: GoogLeNet with one of the inception modules highlighted. Blocks in red are pooling layers, blue blocks indicate convolutional layers, yellow blocks represent Softmax classifiers, and green blocks concatenate the outputs of inception modules. Image source: [78].

In that same year the Visual Geometry Group of the University of Oxford published VGG. VGG also had the ambition to go deeper using convolutions, but did so with a stronger emphasis on consecutive convolutions. VGG relied on consecutive blocks of 3×3 convolutions followed by pooling layers. By using consecutive blocks of 3×3 convolutions the model extracts more refined and non-linear features using fewer parameters when compared to larger filter sizes. It also featured three fully-connected layers at the end of the model. Figure 2.9 displays the 16-layer version of the VGG architecture. VGG notably furthered the belief that depth is important for CNNs and that traditional architectures like AlexNet and LeNet are able to achieve state-of-the-art performance without requiring complex adaptations such as with GoogLeNet. The downside of VGG are its extreme parameter requirements, especially due to its three fully-connected layers.

Fig. 2.9: The VGGNet architecture. It uses blocks of consecutive convolutions followed by pooling layers. Image source: [111].

The next major improvement in architecture design was the ResNet architecture [41], which is able to reach a considerable accuracy whilst using relatively few parameters, a manageable number of operations, and a simple optimization function. ResNet also advanced the number of layers used in deep neural networks considerably and controlled for the vanishing gradient problem that had occurred in previous works. He et al. noticed that despite controlling for overfitting, network accuracy decreased as depth increased. They proposed that stacking identical layers onto a stack of convolutional layers in a traditional architecture (e.g. VGG, AlexNet, LeNet), should theoretically not decrease the accuracy of the network, yet experimentally they noticed that this was the case. This deviation in accuracy was found to be caused by the residuals of the function to be learned, which is the error by misclassification. Traditional network layers could not learn the identity mapping if there were no residuals to correct for as the gradient had to pass through the layer to update earlier layers. The authors proposed to solve this problem with shortcut connections as in Figure 2.10.

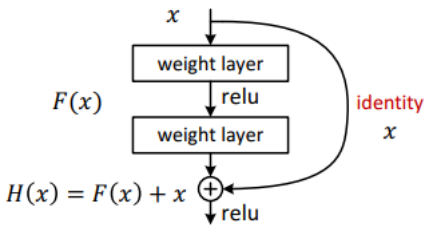


Fig. 2.10: Identity mapping as applied in ResNet models, where F(x) is a module with two 3x3 convolutional layers, BatchNorm, and ReLU activations, x are the incoming feature maps, and H(x) is the function to be learned at the end of the module. Image source: [41].

As a result, if the module has nothing to add to the prediction, it can be deactivated (weights set to zero). During training, this lets $F(x)$ get closer to 0 (no activations) when x gets mapped to $H(x)$. Skip connection allowed models to be trained with greater ease and fewer parameters. It also allowed deeper models to be trained than had previously been the case. Further, ResNet architectures heavily rely on Batch Normalization (BN) layers [49], which in the ResNet architecture are layers that normalize the outputs of a convolutional layer before the activation function. BN layers ensure that weight updates no longer have to adapt to changes in the distribution between layers in a network. They do so by enforcing a unit Gaussian distribution so that input training batches have zero mean and unit variance. As a result, Networks using BN layers can reach convergence rates up to 14 times faster when compared to identical architectures that don't use BN layers [49]. Using BN layers has also empirically shown to reduce the need for regularization methods, thus further increasing robustness against poor initialization.

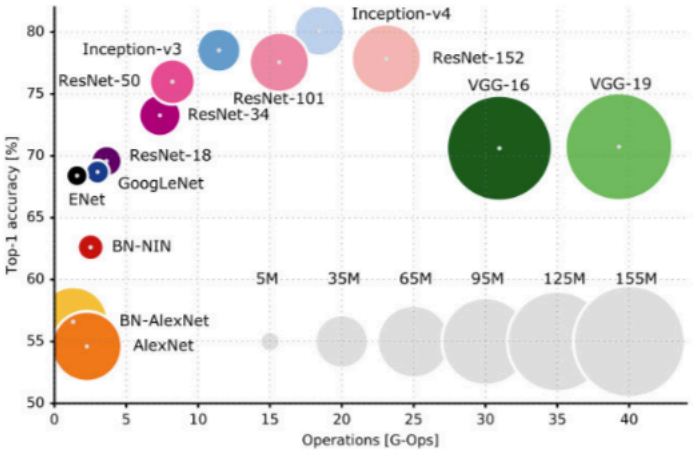


Fig. 2.11: Overview of models according to their top-1 accuracy on the ImageNet dataset, number of parameters, and the amount of performed operations. Image source: [18].

While deeper and more refined architectures have been published in the past years we do not discuss them in this background section. Figure 2.11 gives an overview of the classification accuracy of popular model architectures on the ImageNet dataset [26] versus their number of parameters and the operations required.

2.4.2 Hyperparameter Tuning

Control over the learning process is exerted through *hyperparameter tuning*. A hyperparameter is not a parameter learned by the network, but instead one that is set by the user during the learning process. Choosing the correct hyperparameters is key for training neural networks. For instance, wrongly parameterized networks may attempt to learn too quickly and suffer from overfitting due to lack of regularization. Often, hyperparameters are derived experimentally. The best practices for hyperparameter tuning include searching for parameters on a logarithmic scale, random searching instead of grid searching (searching for combinations using set intervals rather than sampled from a distribution), and starting off with large increments until settling on finer modifications to the parameter in question [61]. A model is trained for a number of epochs by evaluating a set of images in the dataset iteratively. During one epoch, the model thus iteratively learns from images within the dataset until every image has been shown.

The choice of hyperparameters depends on the layers present in a network, as well as the optimization algorithm to be used. The following hyperparameters are relevant for this research:

- Total n epochs: The total amount of times that the model is shown the training dataset. During each epoch, the model is shown every image in the dataset once. The amount of epochs depends on the task, model architecture, and the tuning of other hyperparameters, which may control the speed of convergence and the degree to which the model converges compared to its theoretical maximum potential.

- Batch size: The batch size is the amount of training samples that are shown to the model in an iteration. Using batch sizes above one allows models to use batch normalization layers and will lead to faster processing times as the model can apply the same filters to multiple examples at once. For sufficiently large batch sizes come with a trade-off towards generalization error, which is observed to increase as a result of increasing the batch size [92].

- Initial learning rate: The initial learning rate is the speed at which the model starts to learn using optimization methods such as gradient descent. The learning rate is a multiplicative value applied to the gradient magnitude. Lower learning rates thus means that the gradient signal is weaker, and updates to the network are thus less prominent. It can be interpreted as the maximum speed at which a ball is allowed to roll down a hill.

- Learning rate decay: It is worthwhile to slow down the learning process once the model stops learning from the set learning rate. This is done by reducing the learning rate. This can be interpreted as slowing down the ball rolling down the mountain. If the ball is bouncing back and forth between mountains without slowing down, it helps to slow it down so that it can roll into the ravine below.

- L_2 regularization strength: During evaluation by the loss function, for every weight in the model we apply the term $\frac{1}{2}w\lambda^2$, where w are the weights of a given layer and λ is the regularization strength which can be chosen. This penalty encourages weights to be small, which in turn encourages simpler networks without overly specific filters. As a result, applying regularization reduces the capacity for a model to overfit [56].

Many benchmark models and optimization algorithms are published with suggested ranges and initialization values for use during training, which help to determine which ranges to search in.

2.4.3 Model Fine-tuning

As established, CNNs perform feature extraction through learned filters. While some of these filters may be task-specific, many high-level filters are general filters such as edge detectors which are commonplace between tasks. It is possible to re-use these filters learned on one task to a new task through a method known as *Fine-Tuning*. Model fine-tuning attempts to improve the efficiency of learning a new task by using filters learned from another task. During model fine-tuning the last fully-connected layer is reset along with the classifier, and earlier layers in the network are opened up to receive gradient, and thus to have their filters adjusted [73]. Re-training the entire network has the benefit that it already has learned filters which may simply be fine-tuned to better fit the task at hand. Fine-tuning may also consider re-training only the final fully-connected layer and the classifier whilst freezing the rest of the network, which lets the pre-trained network act as a *fixed feature extractor* [73]. For the purposes of model fine-tuning, programming frameworks for CNNs often supply models with pre-trained weights for popular model architectures.

2.4.4 Interpreting CNN Classifications

A common problem that arises with scene annotation using CNNs is the lack of insight into the triggers for each class, as millions of parameters influence each decision. This makes it difficult to determine whether the model is learning the right representation for the class in question. A particular problem specific to this use case is the lack of data and unique representations, and as a result the model could base its decision-making on the wrong visual cues. For instance, the model might recognize landslides only by cliffs, which are a rare occurrence for the other incident classes. Given the millions of parameters in a network, manually inspecting and inferring filters is an arduous task. Instead, various methodologies have been developed to help interpret and understand the behaviour of CNNs.

Visualization of Filters

With regards to modern deep convolutional neural networks, research performed by Simonyan et al [94] explores two techniques to gain insight into the workings of CNNs. The first proposed method aims to find *Class Model Visualisations* using back-propagation, where instead of optimizing the weights, the optimization algorithm optimizes the input image so that it maximally activates the weights of the network

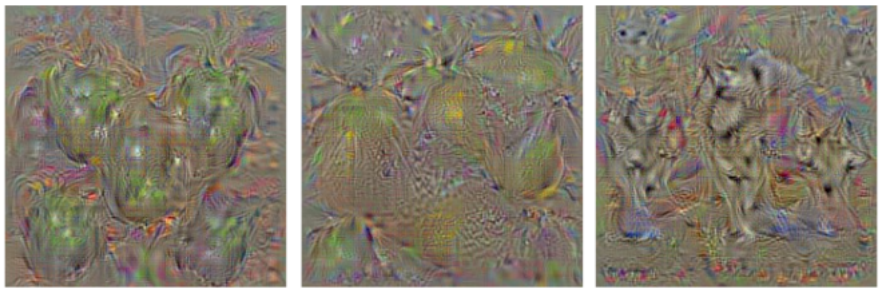


Fig. 2.12: Class model visualisations for a given a target class. Each figure represents an image to which its target class maximally responds. Image source: [94].

with respect to a certain class. This creates a visually-ideal image for a target class which can be seen in Figure 2.12.

The second method proposed by Simonyan et al considers image-specific class saliency. The formulated task is to rank input pixels of a given image on their relative influence in the classification function. The pixels of the resulting images can be interpreted as the pixels which need to be tweaked the least to affect the classification outcome the most. Figure 2.13 provides an example of images and their computed saliency.

A third type of visualization concerns *Class Activation Mapping* (CAM) [116]. Class-activation mapping is based on the observation that convolutional layers are able to perform feature localization within an image without being explicitly being tasked to do so [117]. This characteristic is lost once features are run through the fully-connected layer, which reduces the stack of matrices to a single vector. Intuitively, CAM can be understood as the linear sum of the presence of all visual patterns at different spatial locations within an image. Thus, using CAM may help to understand the visual triggers of a target class within a particular image. The process and result of using CAM is given in Figure 2.14.

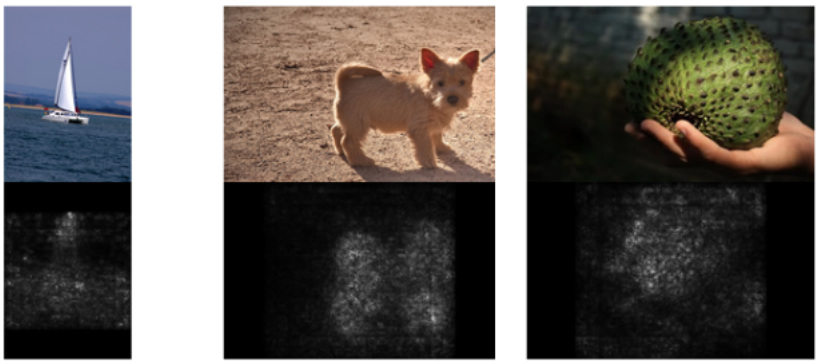


Fig. 2.13: Visual saliency for several images from the ImageNet dataset, where whiter pixels indicate a higher visual saliency. Image source: [94].

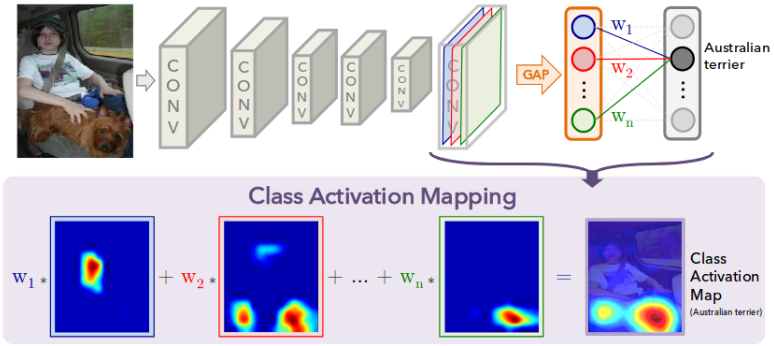


Fig. 2.14: Example CAM class attention for the class *Mastiff*, with yellow-red hues indicating a higher class attention.

Dimensionality Reduction

Beyond inspection of the filters of the model, dimensionality reduction methods can be applied to determine which images the model perceives to be similar. Of these, a method suitable for high-dimensional data is the *T-distributed Stochastic Neighbor Embedding* (t-SNE) method [62]. t-SNE first computes a similarity matrix $S1$ consisting of the Gaussian probabilities that a set of points could be neighbours based on their similarity according to the Euclidean distance between all sets of points their original high-dimensional space. Next, it randomly arranges all data points in a lower-dimension space. In the lower-dimension space it computes a second similarity matrix $S2$, but instead it uses the probability of the Student-t distribution rather than the Gaussian distribution. Through gradient descent it then tries to minimize the Kullbach-Leibler (KL) divergence between $S1$ and $S2$, which is used to determine the location of each point on the lower-dimensional mapping at each successive iteration, minimizing the distance of each point to similar points and maximizing the distance to dissimilar points. In the case of CNNs, t-SNE can be used to determine perceived similarity using the input features of the fully-connected layer. Visually similar images will have similar filters and triggers, and are thus more likely to be clustered by t-SNE.

t-SNE has several parameters that can be adjusted to exert control over the dimension reduction process:

- Iterations: The number of iterations determines how many times the algorithm goes over every datapoint in the dataset to optimize the KL-divergence, and thus to cluster the points together. Iteration should continue at least until the algorithm has converged.

- Learning rate: The learning rate controls the magnitude of updates made to minimize the KL-divergence at every timestep. The expected range for the learning rate is in the order of 10 to 1000 for most tasks [89].

- Perplexity: The perplexity is a measure for information that is defined as 2 to the power of the Shannon entropy. It can be interpreted as a smooth function for the selection of the amount of nearest neighbours. In effect, perplexity determines how many points should be considered when forming clusters. When the perplexity is too high the data tends to clump together and form a singular ball. When it is set too low it fails to find meaningful clusters. The value to use for perplexity should be based on the size of the dataset. Bigger datasets generally require a higher perplexity [106].

When properly configured, t-SNE gives an impression of the structure of the data within the high-dimensional space.

3 Methodology

In this chapter we discuss our choice of methodology and display our settings as applicable. Figure 3.1 gives a sequential overview of the topics discussed in this section.

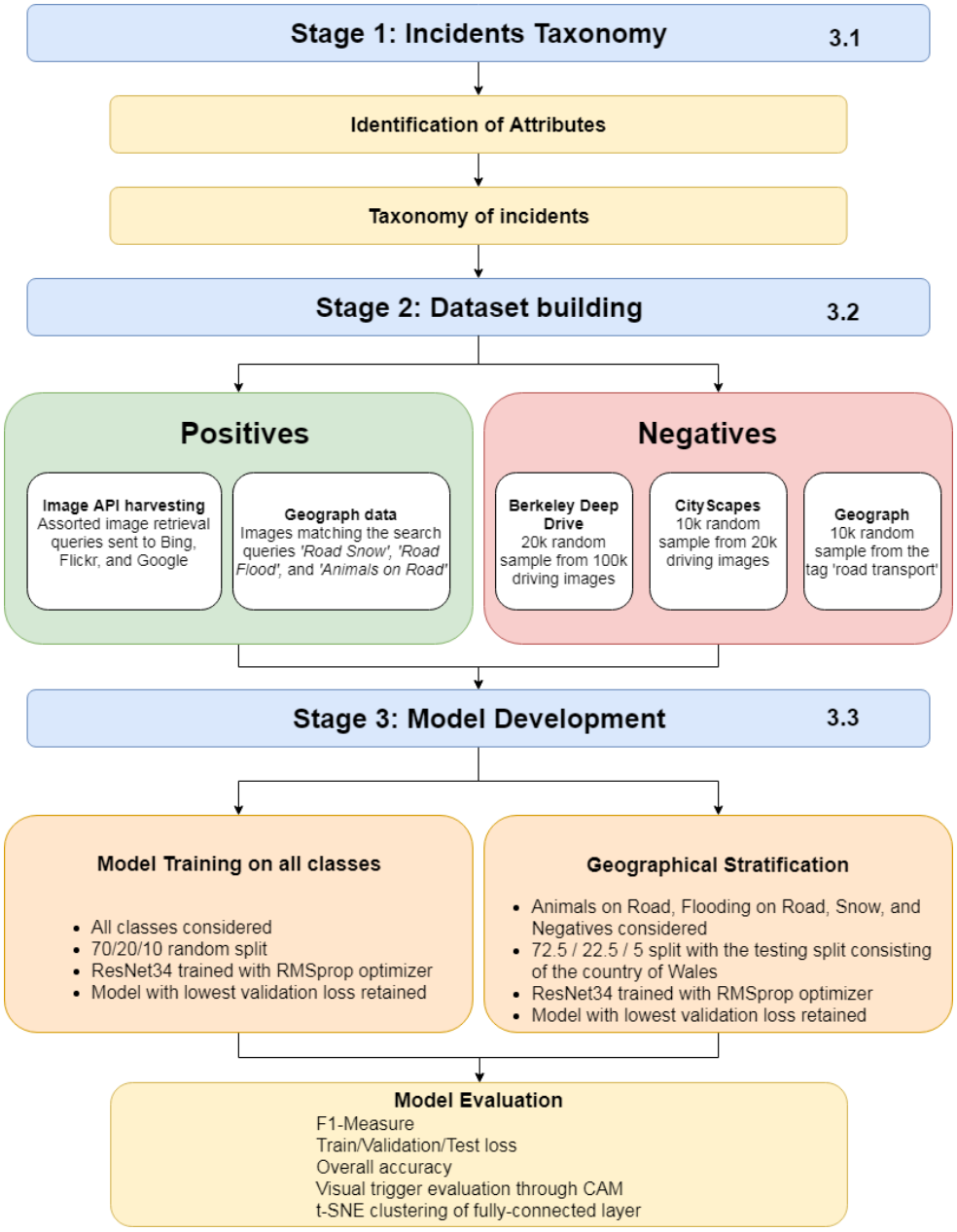


Fig. 3.1: Overview of project methodology and the distinct phases, as well as their outcomes.

3.1 Formalizing Unsigned Physical Incidents

There are many incidents which may affect a road network, and many incidents are indeed related. For instance, snowy and inundated roads are distinct phenomena, but related incidents as they both pertain to natural events which affect the road surface. In order to consider the breadth of possible incidents we therefore propose the use of a taxonomy to establish semantic groupings which can be adapted to local circumstances as needed.

Using similarity in attributes to relate incidents to one another provides the benefit of groupings. Groupings can be used to classify related events that may be hard to classify as individual incidents, for instance due to their rarity. In order to determine the structure of the taxonomy we have performed a Formal Concept Analysis (FCA) [33] in order to uncover attributes and attribute groupings relating to incidents. We iteratively refined and grouped attributes until they provided binary semantic groupings at various levels. These semantic groupings allow for the option to classify incidents at a higher level. This allows for the classifier to be assessed on which semantic groupings it can recognize, as well as to determine which groupings can be assessed at a more complex semantic level in future work. From the FCA we identified two aspects that were commonly shared between incidents. The first grouping is the most likely cause of an incident. We identify that incidents can be caused by man-made or natural causes. For instance, a car crash involving two cars is most often the cause of driver error or a mechanical failure, both of which find their root in a human failure. Flooding on the other hand is most likely caused by natural causes. The second grouping is whether the incident is a well-defined discrete (set of) object or a continuous field, which we refer to as a *cover*. Flooding is a continuous phenomenon, since there is no discrete delineation of the incident. On the other hand, a flock of sheep can be counted, and thus be considered a discrete incident, which we refer to as *objects*. We display how this may be formalized in figure 3.2, which is the resulting taxonomy of incidents that we maintain in this research.

The taxonomy has been designed with several considerations in mind.

- While a glitch in a traffic system may cause panic on the street, it is hard if not impossible to detect this incident in imagery. Hence, the inclusion of digital incidents is beyond the scope of this research and not considered. In a full hierarchy of possible incidents it will be on the same level as physical incidents.

- We consider attributes for incidents by their perceivable cause. For instance, it is possible that a tree was purposely cut to block off a road, making it a man-made adversarial incident. However, such an incident is often hard to distinguish from natural treefall, which is far more abundant in occurrence.

- Signed physical incidents are not considered in this research as the detection of signage such as road signs are broadly researched using computer vision techniques. With sufficient research progress the detection of road signs may soon cover for the detection signed incidents and events such as roadworks and parades.

The lowest-level labels refer to individual incidents that may belong to the specified groupings. In this research we consider 8 lowest-level labels for classification. We chose a variety of incidents that can easily be semantically described for the purposes of searching for images and for which we have a reasonable indication that

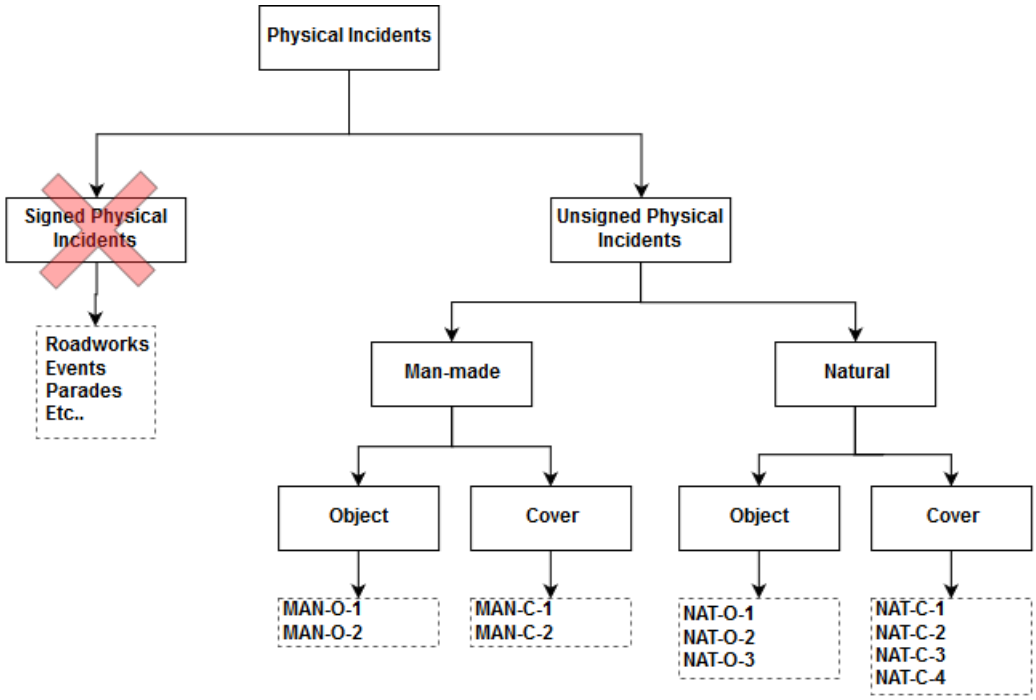


Fig. 3.2: Taxonomy of incidents and their Semantic groupings

we can collect hundreds of images in the research period. We also tried to cover the spectrum of combinations in objects and surfaces. We use the following working definitions during the research:

Animal on Road: Any animal, both living and dead, situated on or within close proximity of the driving surface

Collapse: A major break-up of the driving surface which would be too big for common motor vehicles to drive across without incurring damage

Fire: An uncontrolled and active fire anywhere in the image that may affect the driving conditions immediately or when left uncontrolled

Flooding on Road: A (section of) driving surface that is submerged in a cover of water puddles such that it causes drivers to change their driving behaviour

Landslide: A cover of dirt, rocks, or natural debris originating from a raised surface, which has settled on the driving surface

Snow on Road: Any amount of snow on the driving surface such that it could cause drivers to change their driving behaviour

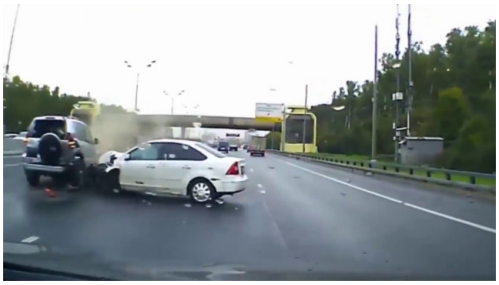
Treefall: A tree, trunk, or sizable branch leaning over or lying on the driving surface in such a way that it would obstruct traffic

Vehicle Crash: Any visible collision between one or more motor vehicles, or a motor vehicle collision with an object in the environment, such as a tree

The chosen classes and their attributes are given in table 3.1. We list one prototypical example for each image class in figure 3.3 to illustrate how incidents look operationally.

Incident	code	Man-made	Natural	Object	Surface
Vehicle Crash	MAN-O-1	X		X	
Road Collapse	MAN-S-1	X			X
Fire	MAN-S-2	X			X
Animal on Road	NAT-O-1		X	X	
Treefall	NAT-O-2		X	X	
Snowy Road	NAT-S-1		X		X
Flooded Road	NAT-S-2		X		X
Landslide	NAT-S-3		X		X

Tab. 3.1: Positive unsigned incidents under consideration during this project.



(a) Vehicle Crash [12]

(b) Road Collapse [8]

(c) Animal on Road [1]

(d) Treefall [9]

(e) Snow on Road[95]

(f) Flooding [79]

(g) Landslide [105]

(h) Fire [102]

Fig. 3.3: Prototypical examples of the incident classes covered in this research.

3.2 Dataset Creation

This section discusses the process for setting up the training dataset. The data for this project consists of imagery collected from various web sources and benchmark datasets. In this chapter we first describe the constraints to which suitable images are subjected, followed by an overview of the sources and compositions used for the positives and negatives of the dataset.

3.2.1 Imagery Constraints

The input data for this project is imagery of various incidents occurring on the road network. However, it is necessary to constrain the input imagery to certain rules in order to filter the imagery to relevant examples. The proposed constraints cannot be automatically tested, and thus the human reviewer has to assess each images' suitability manually. We maintain the following guidelines to determine whether an image is suitable:

- Viewpoint roll
  There are only various possible angles in which a camera is likely to be fitted on a vehicle. Under normal operation it is unlikely that a vehicle-mounted camera is ever rotated beyond a couple of degrees in either direction.

- Viewpoint height
  Given the range of vehicle ride height for various vehicles, we consider images taken between 50cm (sportscars) and 4 meters (trucks) height above the surface for inclusion.

- Viewpoint pitch
  The viewpoint should be forward or slightly-downward facing. Images should have a maximum allowable pitch of 15 degrees either up or down.

- Viewpoint yaw
  The viewpoint should face the road network as much as possible. For instance, an image taken from on the road, but having very little road in view of the image is most likely irrelevant to the assessment of road network conditions. As such, the yaw should ideally be less than 30 degrees in either direction.

- Lateral offset
  The viewpoint should be positioned on or close to the street in order to be representative of the vehicle's spatial position on the street. The maximum

allowable camera viewpoint should be the edge of the sidewalk (e.g. to simulate a parked car).

- Incident location
  The incident in an image should be relevant to the driving conditions. For instance, snow on the side of the road is not an incident if the driving lane itself is cleared of snow. On the contrary, a car crash with a car standing on the side of the road is still relevant for driving conditions of other cars of the road as network accessibility may be reduced during cleanup operations.

These guidelines should constrain all of the imagery used in the research to only contain imagery which is relevant for the incident affecting the road. That said, these guidelines are not definitive, and ultimately it is up to the human interpreter to determine whether an image is relevant or not. Depending on the data availability for each class we may relax these constraints. For every image we only use a single image label. That is, only one class exists per image. For instance, images that contain both a crash and fire are not included.

3.2.2 True-Positives Dataset

In this section we discuss our procedure for generating a set of images that contain incidents, which we refer to as the *true-positives* dataset. We first discuss our approach to collecting images from web sources before considering image selection, storage and the generation of a labeled dataset consisting of normal driving conditions, which we refer to as *true-negatives*.

Web Retrieval Approach
Given the high amount of data required to train CNNs there exists a need to build a large dataset. Furthermore, images have to be diverse in nature to represent the many possible conditions in which incidents may occur. For instance, snowfall can occur anywhere and the visual cues in a city may be different than those of a snowy forest road. In order to maximize the amount of data that can be gathered during the project we maintain a retrieval strategy inspired by previous work on the ImageNet [26] and LSUN [110] datasets.

Image Searching
Image searching is done by using the APIs of three search engines, as well as using images from the Geograph UK project [35]. We send prepared queries to the Google Custom Search API [37], the Bing Web Search API [67], and the Flickr Image Search API [109]. For each API, the user prepares a query and sends it to the API endpoint, which returns a list of image URLs matching the query.

The Google Custom Search API lets users query for images through a custom search engine, which is a customizable instance of the regular Google search algorithm. We set up a custom search engine with the only modification to the standard settings being the filtering of sites. We filter out the top-level domains of 13 stock photo suppliers: *dreamstime.com, depositphotos.com, pixoto.com, bigstockphoto.com, istockphoto.com, fotolia.com, colourbox.com, 123rf.com, fotosearch.com, alamy.com, gettyimages.com,* and *shutterstock.com*. This reduces the amount of copyrighted stock photos appearing in the search, and thus improve the returns from the search engine. Bing and Flickr's API have no customizable settings with relation to the search engine itself. Customization is only done through prepared queries.

For the Geograph UK project we contacted the organization and requested a set of images matching the following search terms: *road snow* for images with snow, *road flooding* for images with floodings, *cows/ducks/sheep on road* for images with animals.

Image Retrieval and Storage
Searching using an API allows for parameters and options to be set. We send the API query with the following parameters specified for each API:

- **Google Custom Search**[1]
  imgType: *photo*
  googlehost: *google.uk* (Specifies which Google host should handle the query)
  gl: *uk* (Sets the geolocation of the query)
  cx: *013675800614641398741:wwg9y3xxkj0* (See previous section; specifies the custom search engine instance to use)
  filter: *1* (Filter duplicates within the query)
  imageSize: *medium* (Return images which have a size of 'medium' (500 pixels))

- **Bing Image Search**[2]
  imageType: *photo*
  offset: *0* (starting index of matching images)
  count: *100* (Number of images per query)

- **Flickr Image Search**[3]
  tag_mode: *all* (specifies to search for images that contain the intersection ('and') of all tags)
  format: *json* (The format in which the response is sent)

[1]https://developers.google.com/custom-search/json-api/v1/reference/cse/list
[2]https://docs.microsoft.com/en-us/rest/api/cognitiveservices/bing-images-api-v7-reference
[3]https://www.flickr.com/services/api/flickr.photos.search.html

media: *photos*
per_page: *100* (amount of images per query)
page: $per\_page * 100$ (index of images, determined by per_page)
sort: *relevance* (returns most relevant images)
nojsoncallback: *1* (gives a raw JSON return without function wrappers)

All prepared queries have to be sent with an API key. Google and Flickr provide free API keys for registered users, while Bing Image Search is a paid service with a free trail, which we make use of during this research effort. Google limits the query rate of free users to 100 unique queries per day, thus limiting the total amount of images retrieved per day to 1,000. Querying is thus done over the span of weeks. Each query sent returns a JSON file with the URLs of all images in the query. We download each returned image and retain the returned JSON file for metadata purposes. Bing and Flickr-derived images provide a unique image ID for metadata-retrieval purposes, while Google does not. We store Flickr and Bing images with their relevant image ID, and we generate a 10-character random filename for Google images.

Query Expansion

While all of the listed search engines implicitly perform query expansion (broadening the results gathered by a query by considering heuristics such as synonyms and spelling) when retrieving search results, it is still worthwhile to searching using explicit synonyms. Doing so returns a wider variety of images and expands the search in breadth rather than in depth, as the quality of depth-wise searching typically had a noticeably drop in quality after the first few pages. Furthermore, breadth-wise searching has the added benefit of displaying various possible representations of the incident in question, such as snowfall in a forest, countryside, or in a city. In this research we used prominent synonyms with varying geographic implications. For instance, we used four variants of road types during the querying process, namely *street, highway, road, and route*. Streets and roads are typically used in (sub)urban scenes, while highways and routes are more often used to denote roads outside of cityscapes. We also consider various other terms for the incident in question, such as vehicle types when considering crashes. A full list of expanded queries is available in Appendix A.

Multilingual Queries

Beyond the English language we include queries of various other languages native or skillful to colleagues. By doing so we aim to retrieve images that are not returned through English queries, as well as to capture a greater geographical diversity as a result of the inclusion of resources not covered by the English language. The following languages have been used for querying: *Dutch, Slovak, Mandarin, Croatian,*

and Farsi. To do so, we asked colleagues to translate the subject into a query that they would use to find images in their native or proficient language. We provide an example in English and ask colleagues to translate it. To save time on the cleaning process, we first manually enter each query at the three API hosts to inspect whether the query yields images of the incidents in question. Approved queries are then sent to the API so that images may be harvested. We list the table with multilingual queries in appendix A.

Selecting and Storing Images
Before image selection is performed we remove duplicates from the returned images. We check each queried image within an incident class for exact equivalence to every other image in the incident class. This process is meant to eliminate redundancy during the selection step and to reduce the chance that duplicates are added to the dataset. Since Bing returns more metadata than Google and Flickr we remove identified duplicates from the images returned by Google and Flickr. During image cleaning, for every returned image we manually determine whether to include the image to the dataset or not based on the criteria presented in 3.2.1. We assess images through a self-written terminal tool where images are resized to 500x500 pixels and assessed for suitability. To do so, resizing is done using *Nearest Neighbour* interpolation so that images retain their original representation as much as possible. Selecting images is done only by the author. While this adds a significant amount of workload to the research, it ensures that there is stability in the semantic definition of the dataset. We consider this to be integral to the research, as the primary intent is to determine whether incidents are recognizable from image data. It also adds to the discussion on the specifics of the dataset as its exact content is known.

3.2.3 True-Negative Dataset

Since this research is concerned with recognizing incidents, a dataset of negative examples has to be sampled. For this purpose we sample from various sources to cover the preconditions of the research:

- **Berkeley Deep Drive (20k)**
  we sample 20k images from the 100k images subset of the Berkeley Deep Drive (BDD) dataset [90]. This dataset is notable for its variety in scenes and weather conditions. The inclusion of scenes containing wet and snowy conditions makes this dataset especially relevant to help distinguish between disruptive and non-disruptive weather conditions. It also contains variations in rotation and angle similar to the incidents dataset. The scenes are recorded in a variety of cities in the United States and contain the only the 10th frame of each video in their 100k videos dataset. We sample 20k images from this

dataset as it captures the widest variety of driving conditions of the negatives datasets such as weather conditions, geographic diversity (within the United States), day/nighttime, and complications such as reflections of the dashboard.

- **Cityscapes (10k)**
  To further improve the geographic diversity of the dataset, we sample images from the Cityscapes dataset [24]. This dataset covers a variety of street scenes of German cities and contains every 20th frame of 30fps video sequences. While it contains a low variety of weather and camera conditions, the driving conditions in the dataset are distinctly more European than the BDD dataset. We include 10k images from the Cityscapes dataset to improve the geographic diversity of the negatives class.

- **Geograph (10k)**
  Finally, we use a random sample of 10k images tagged as *'road transport'* from the Geograph project [34]. The dataset contains unfiltered stills covered by the tag, and therefore reflects all sorts of photos, at times even irrelevant to driving conditions in general. While most photos are taken from a viewpoint similar to BDD and Cityscapes, a number of images also contain odd angles and targets (e.g. streams or pastures). We include this dataset to offset the strong urban focus of the aforementioned benchmark datasets, as well as to counteract potential overfitting on the difference in viewpoint rotation, angle, and orientation in the incidents dataset. We retain 10k images from this dataset to ensure the inclusion of landscape images, which are frequent in the positives dataset. The 10k images are a random set returned from the search term *road transport*.

- **Geograph Snow Negatives (200 images)**
  As an experiment we collect approximately 200 images of boundary cases for the *snow* class of the Geograph data to determine whether it is able to pick out negative cases based on the relevant attributes, which is snow *on top of* the driving surface. These images are selected when filtering for positives in the class *snow*. Figure 3.4 gives an example of a positive image with snow versus a negative image with snow.

Pre-processing

Images from the BDD dataset often contain elements from the ego vehicle in the image itself such as the dashboard and the bonnet. To reduce the chance of overfitting the negatives dataset onto irrelevant visual cues we crop out the bottom 25% of all

Fig. 3.4: On the left: a true-positive image of snow which contains snow on the driving surface [101]. On the right: A true-negative image of snow which does not contain snow on the driving surface [36].

images in the dataset. To compensate for the change in aspect ratio we crop 12,5% from both the left and right side of the image, thus retaining the aspect ratio.

Images are resized to 224x224 to match the input size of ResNet. We do not consider cropping to be viable on this dataset as it may risk cropping out the incident, since incidents in images are frequently small and off-center

3.2.4 Data Management Strategy

We maintain a data management strategy to ensure that the dataset can be used for future research. During the research we retain all collected images until the end of the research effort on an external hard drive and regularly back it up on the Google Drive service of the university. Images are stored in a SQLite database with their relative file path from the top of the data folder, their calculated split, and their class. SQLite supports querying and allows for a more structured file storage than plain text storage methods, while providing the benefit that resulting databases are small. Furthermore, lightweight readers are available to view records and perform elementary queries. For non-commercial use we distribute a zip of the images that are used to train the complete model in the following structure: `split/class/images`. The folder thus consists of three `split` parent folders, and eight `class` folders for each parent folder. We will not supply the negatives images directly as BDD and Cityscape's licenses do not allow re-sharing of the dataset. Researchers can instead re-construct the negatives dataset with the scripts generated in this research. We will make the Geograph negatives dataset available with a copyrights file to ensure Creative Commons compliance, as per the Geograph project's request. We will create a SQLite database containing the image id, class, split, and URL to the original image so that users can reconstruct the positives of the dataset themselves.

3.3 Incident Recognition

In this chapter we discuss our set-up for the incident recognition models, as well as the settings for the visualization and insight methods.

3.3.1 Model

To make the most use of the limited amount of data we apply fine-tuning onto a ResNet-34 [41] model which has previously been trained on the ImageNet dataset. To account for the imbalance in the number of positives and negatives and to limit overfitting onto specific classes, we perform weight updates relative to the inverse fraction of a class' images compared to to the total set. For instance, if the negatives class contains 80% of the images, each weight update resulting from these images is multiplied by 0.2, thus reducing the influence of the negatives class at each update.

We evaluate our model using two metrics. Firstly we present the average top-1 accuracy (the model's *best guess* for a given image across all n images within a class, not taking into account false positives) for all classes. We then expound the accuracy by considering the confusion matrix of all classes. We use the F1-measure Van Rijsbergen [103] to consider the model's accuracy with regards to the inclusion of false positives, which is given as follows:

$$F_1 = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.1}$$

Here, the precision is measured as the number of images correctly predicted to belong to a class divided by the number of all images predicted to belong to that class. Recall is the number of correctly predicted images for a given class divided by the number of images that should have been predicted to be positive. The *F1* measure ranges between 0 and 1, where 1 indicates perfect precision and recall, and 0 means no precision or recall at all. The F1-measure represents the harmonic mean of the precision and the recall and gives a good indication whether a model is balanced in its predictions. For instance, if a model always predicts negatives, it may reach an overall accuracy of 100% for the negatives (all negative images classified as such), but in reality its predictions are baseless. The average F1-measure for this class would be lower, as it considers the amount of images that should have been returned versus the total number of returned images, and thus punishes the model for overzealously classifying images to a given class.

3.3.2 Optimization

We optimize the model using the RMSprop optimization algorithm [44, s.29] without applying momentum. RMSprop is part of the adaptive family of optimizers, which are conceptualized to deal with common problems occurring with stochastic gradient descent. We chose the RMSprop algorithm for one particularly favourable property: Gradient updates are performed by keeping track of recent gradient magnitudes and subsequently subtracting the running average from the parameter, which has a smoothing effect on gradient updates. We do not consider momentum during optimization as the implementation of momentum in the coding environment did not work. Optimization is only performed during the training stage of each epoch.

3.3.3 Training Settings

Training settings are derived experimentally. We initialized the model with a 0.001 learning rate and observed the convergence pattern. From there we reduced the initial loss by a factor 10 and set up our decay schedule to decay when the model would stop improving. We did not tune other hyperparameters as the observed accuracy was already satisfactory after convergence. We thus initialize and train the model using the following parameters:

- **Batch size**: 10 images per batch
  We use the maximum number of images per mini-batch that the GPU on the implementation environment allows.
- **Total number of epochs**: 50
- **Initial learning rate**: 0.0001
- **Learning rate decay schedule**: 10, 30, 40
  We don't decay the learning rate between the 10th and the 20th epoch as the model continued to converge quickly during experimentation runs.
- **L2 regularization strength**: 0.0001 (ResNet default)

To artificially improve the robustness of the trained model we apply various random augmentations to the training data, with the probability of occurrence set to 50% where applicable:

- **Random horizontal flip**
  Horizontally-flipped driving images still pertain to incidents on the road and helps to train filters that are less sensitive to a particular driving side.

- **Random grayscale transform**
  Reducing the image colour information to grey tones may help to reduce dependency on colour-sensitive information, which is relevant for classes such as snow and fire.

- **Random rotation up to 5 degrees in either direction**
  Small variations in rotation can help to train filters that are less sensitive to variations in rotation.

- **Jittering hue/brightness/contrast/saturation up to a factor of 0.05**
  Small alterations to the image may mimic common variations in image quality, and thus make filters more robust against lower-quality images.

Augmentations and values for the augmentations are applied with understanding to the task at hand. For instance, it is unlikely that a vehicle-mounted camera will be rotated beyond a couple of degrees, nor is it expected to see severe colour mutations. Lastly, we normalize all images passed to the model to the mean and standard deviation of the training subset to improve numerical stability. Normalizing to the mean and standard deviation acts as a means to ensure that the input images always follow the same distribution, which stabilizes gradient backpropagation as it no longer sends a gradient signal that varies by the distribution of the input images. During training we apply all augmentations while we only apply normalization during validation and testing.

Network training is performed using PyTorch version 0.3 running on Python version 3.6. The model is trained on the freely-available Google Colaboratory platform [37]. Colaboratory is a notebook environment similar to Jupyter Notebooks, but it allows for free access to a NVIDIA (memory-limited) K80 GPU on which models can be trained on for up to 12 hours at a time. We restart the optimizer to decay the learning rate as Google Colaboratory does not facilitate scheduled learning rate decay during the learning process itself. It is not known whether this is a bug or by design.

3.3.4 Interpretation Methods

In this research we apply Class Activation Mapping [116] (CAM) to the average-pooled final convolutional layer of the ResNet model to determine which visual cues the model responds to. We do so by assessing outliers which we detect using t-SNE dimensionality reduction of the inputs to the fully-connected layer of the ResNet model. For the CAM method we use an existing implementation provided by the authors of the CAM method intended for ResNet models in PyTorch [115]. t-SNE dimensionality reduction is performed using the SciKit Learn [89] implementation.

We use t-SNE dimension reduction to determine which images are visually similar and to detect unusual outliers such as an incorrect classification in the middle of a cluster, as well as to get a sense for the general underlying structure of the prediction function. To do so, we test a variety of perplexity settings. We use the heuristics proposed in [106] to tune the algorithm and interpret their resulting plots.

- **Step count**
  We run the model a total of 1,000 iterations to ensure convergence.

- **Learning Rate**
  We use a learning rate of 500 to ensure a stable convergence. After testing we assessed that it leads to a fast convergence while not sacrificing the quality of the plot.

- **Perplexity**
  Perplexity is tested at rates of 20, 50, 200, and 250 to assess the effect of cluster tightness and to test whether the geometrical structure remains constant.

We then visually assess the patterns visible in the most expressive t-SNE plot and discuss remarkable mis-classifications using CAM.

3.4 Geographical Stratification Set-Up

In order to assess whether incidents can be learned independently from their environment we run an experiment using a geographical stratification. We run an experiment using the three incident classes present in the Geograph data; *Animals*, *Flooding*, and *Snow*, as well as considering *negatives*. We only use these three classes and the full dataset of negatives as the images retrieved from the Geograph project have reliable geotags that are situated in the United Kingdom or Ireland and as such they can be stratified to a region of these two countries. We geographically stratify images based on their location. Images within England, Scotland, or Ireland are included in the training or validation dataset, while images situated in Wales are used in the holdout dataset. In doing so we effectively split the Geograph data to a 72.5/22.5/5% split, with the 5% holdout data situated in Wales so that we can test the trained model performance on unseen data from a new geographical region. Figure 3.5 displays the geographically stratified positive data-points.

During training and validation of the geographically stratified dataset we include the harvested and multilingual data. 75% of the harvested and multilingual data in each of the three positive incident classes is added to the training dataset, while the remaining 25% data is distributed to the validation dataset. While it is possible that some of the scraped data is situated in Wales, we don't suspect that this inclusion will cause significant geographical correlation. The ratio of Geograph to non-Geograph data in this class is greater than 2:1 for all three classes, so the vast majority of all images are accounted for. As such, it is very unlikely that geographical correlation will occur as a result of including multilingual and harvested images. Model training will occur with the same parameters as the model assessing all incident classes, as the chosen hyperparameters were observed to lead to good convergence.

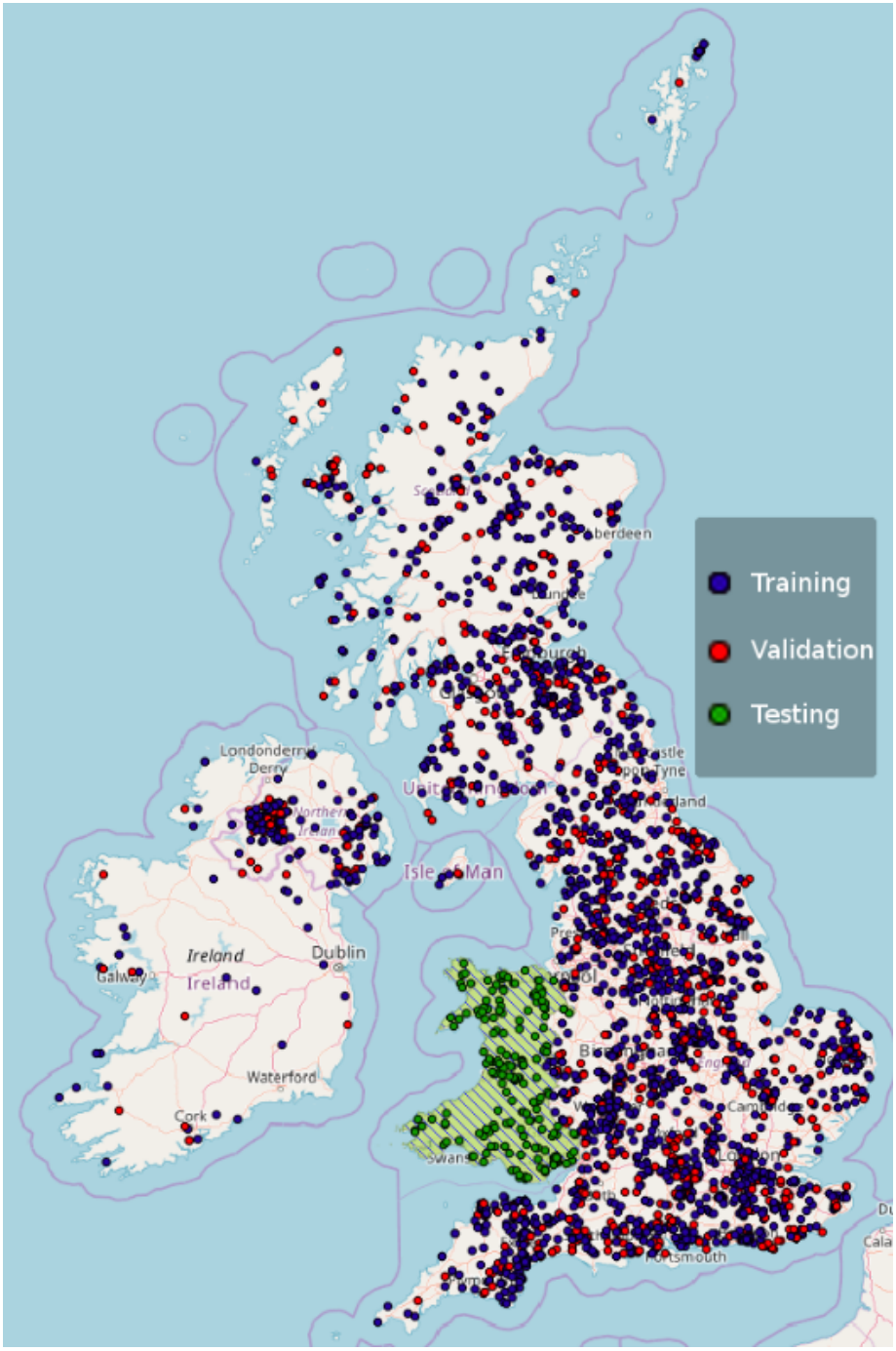


Fig. 3.5: Overview of positive data-points as stratified during the geographical validation. Negative data-points have been stratified in the same manner. Highlighted in green is the country of Wales, which is the unseen geographical region to test in.

4 Results

This section discusses the results that have been derived for the research. Firstly we present the created dataset, and secondly we present our results for the classification. Where applicable, we refer to the dataset covering all eight incident classes as the *complete dataset*.

4.1 Data Collection Results

In this section we discuss the images that have been collected during the research. Appendix A contains the queries performed for the harvested and multilingual queries.

Using query construction by using synonym combinations, we performed 118 queries which retrieved 40,063 images, of which 5,844 were included in the dataset. We retain 2,439 images from Google, 2,742 from Bing, and 663 from Flickr. It is important to note that duplication filtering works in favour of the total amount of images retained from Bing. We did not track the total amount of duplicates removed. Figure 4.1 lists the distribution of images retained per class along with their sources. As can be seen, Google and Bing provide a superior rate of correct images when compared to Flickr, while being highly similar in the amount of correctly returned images per class.

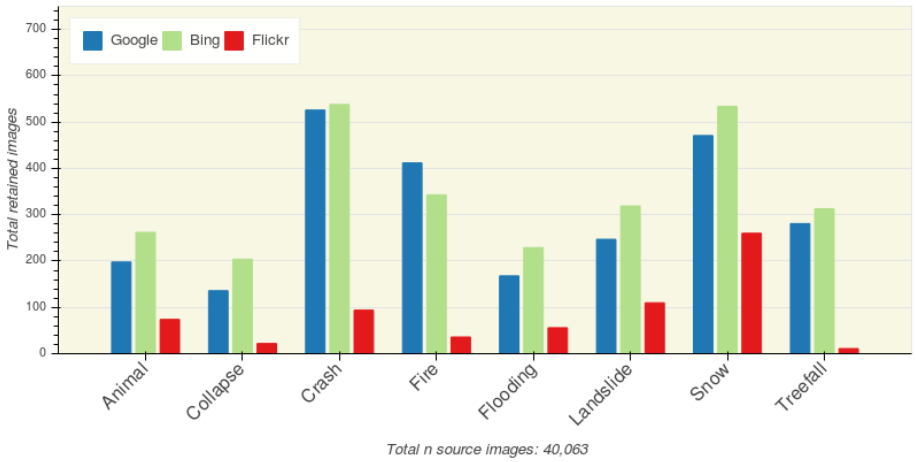


Fig. 4.1: Overview of images per class as derived from each source using the harvesting queries.

By translating representative queries into various non-English languages, we performed 63 queries which retrieved 12,846 images, of which 1,641 were included in the dataset. We retain 762 images from Google, 804 from Bing, and 74 from Flickr. Figure 4.2 displays the distribution of images retained per class by source.

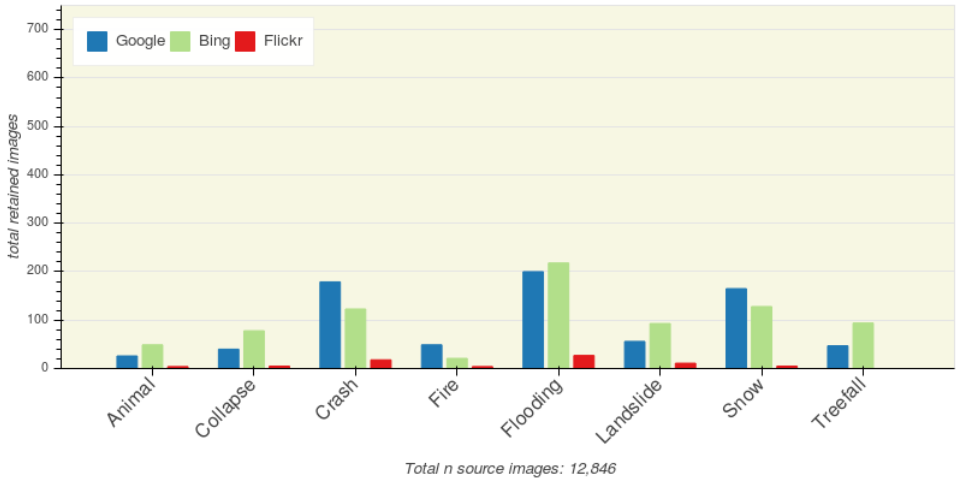


Fig. 4.2: Overview of images per class as derived from each source using the multilingual queries.

In table 4.1 we give an overview of positive collected images as gathered from harvesting queries, multilingual queries, or by cleaning Geograph images.

Incident	Harvested	Multilingual	Geograph	Total
Animal on road	534	79	708	1,321
Collapse	362	123	6	491
Crash	1,158	320	-	1,478
Fire	791	74	-	865
Flooding	453	446	1,257	2,156
Landslide	676	149	-	825
Snow	1,265	304	3,174	4,743
Treefall	605	146	-	751
Total	5,844	1,641	5,145	12,630

Tab. 4.1: Total amount of images per class as collected by gathering type.

40,221 images have been included in the negatives dataset in total, of which 20,000 from Berkeley Deep Drive, 10,000 from CityScapes, 9,981 from Geograph, and 240 negative boundary cases of snow gathered during the cleanup of the Geograph positives dataset.

4.2 Model Classification Performance

Here we present the results for both classification cases, starting with the full dataset.

4.2.1 Complete Dataset Performance

Training was concluded after 50 epochs and we retained the model with the lowest validation loss, which occurred at epoch 37. Table 4.2 displays the final accuracy and F1-score derived for each phase on the best model. A near-perfect accuracy is achieved during training, with the validation and testing set displaying consistently high accuracies with a lower F1-measure, indicating some degree of classification confusion. Remarkably, the model performs better during testing than during validation. The confusion matrices for the training, validation, and testing phase are given in tables 4.3, 4.4, and 4.5 respectively, which displays the expected trend that most misclassifications pertain to the negatives class. Figure 4.3 displays the loss for the model at every epoch during training, while Figure 4.4 displays the accuracy of the model at every epoch. As can be seen, training loss and accuracy improved steadily while the validation loss and accuracy display an erratic convergence pattern until the learning rate is sufficiently decayed after the 30th epoch.

Metric	**Training**	**Validation**	**Testing**
Accuracy	99.49%	96.31%	97.15%
Avg. Unweighted F1-score	0.9403	0.9054	0.8909
Loss	0.02149	0.2135	0.1761

Tab. 4.2: Classification performance of the best model trained on the complete dataset

Predicted / **True**										**F1-Measure**	**Top-1 Accuracy**
Animal on Road	872	1	0	0	0	0	0	1	15	0.9842	98.08%
Road Collapse	0	350	0	0	0	0	0	0	1	0.9957	99.72%
Vehicle Crash	0	0	984	0	0	1	0	0	11	0.9904	98.79%
Fire	0	0	0	590	0	0	0	0	2	0.9966	99.66%
Flooded Road	0	1	0	0	1470	0	0	0	32	0.9840	97.80%
Landslide	2	0	1	0	1	562	3	0	5	0.9860	97.91%
Treefall	0	0	0	0	0	3	522	0	5	0.9887	99.87%
Snow on Road	1	0	0	0	2	0	0	3238	59	0.9891	98.12%
Negative	8	0	6	2	12	0	1	8	27730	0.9970	98.49%

Tab. 4.3: Training split confusion matrix (n=36,502) of the best model trained on the complete dataset. The x-axis represents the true class and the y-axis represents the predicted class.

Predicted / True										F1-Measure	Top-1 Accuracy
Animal on Road	237	1	3	0	1	6	1	0	31	0.8360	84.64%
Road Collapse	0	71	1	0	3	3	0	0	12	0.8160	78.89%
Vehicle Crash	0	0	283	2	4	6	2	1	10	0.9056	91.88%
Fire	1	2	0	174	2	0	1	2	6	0.9560	93.55%
Flooded Road	3	4	3	0	396	4	0	8	44	0.8637	85.71%
Landslide	1	3	4	0	2	151	2	0	12	0.8412	86.29%
Treefall	1	0	1	0	1	9	129	0	7	0.8571	87.16%
Snow on Road	1	1	3	0	3	0	0	988	42	0.9611	95.18%
Negative	43	2	19	0	43	5	18	19	8248	0.9814	98.22%

Tab. 4.4: Validation split confusion matrix (n=11,086) of the best model trained on the complete dataset. The x-axis represents the true class and the y-axis represents the predicted class.

Predicted / True										F1-Measure	Top-1 Accuracy
Animal on Road	129	0	1	0	2	1	0	1	1	0.9021	95.56%
Road Collapse	0	50	1	0	0	0	0	0	3	0.9174	92.59%
Vehicle Crash	1	0	155	0	0	0	1	1	2	0.9394	96.88%
Fire	0	0	0	97	0	1	0	0	2	0.9848	97.00%
Flooded Road	0	1	0	0	188	0	1	2	20	0.8806	88.68%
Landslide	0	1	0	0	0	65	1	0	3	0.9028	92.86%
Treefall	0	0	1	0	1	2	67	1	1	0.9241	91.78%
Snow on Road	2	0	0	0	3	0	0	468	14	0.9689	96.10%
Negative	19	3	12	0	21	6	2	6	3894	0.9854	98.26%

Tab. 4.5: Testing split confusion matrix (n=5,263) of the best model trained on the complete dataset. The x-axis represents the true class and the y-axis represents the predicted class.

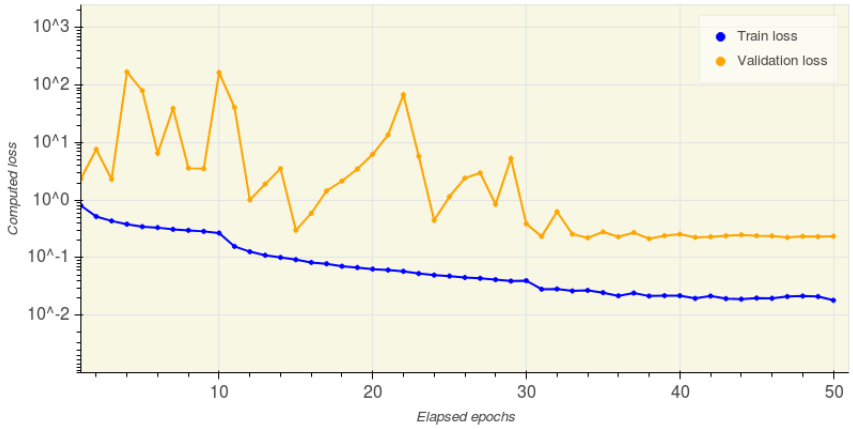


Fig. 4.3: Loss curve of the model trained on the complete dataset. The y-axis represents the loss incurred at each epoch (lower=better), while the x-axis represents the epoch at which the loss occurred. A steady decrease is indicative of a model that is converging well.

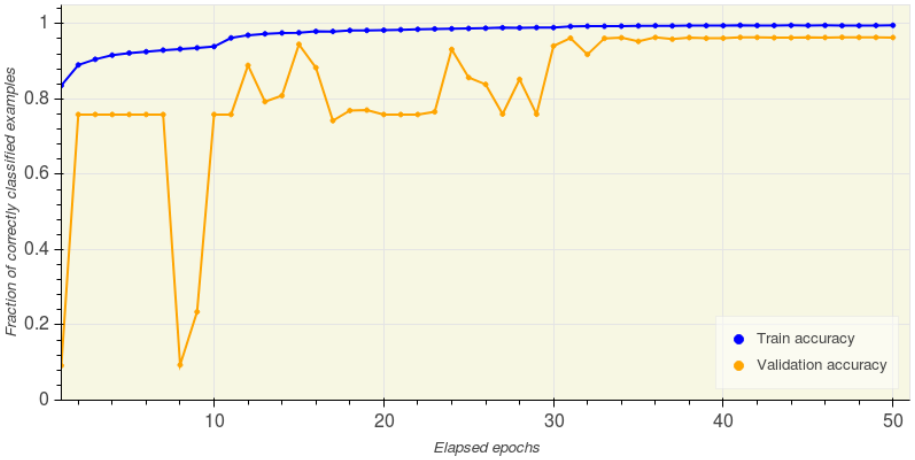


Fig. 4.4: Accuracy curve of the model trained on the complete dataset. The y-axis represents the accuracy rate at each epoch, while the x-axis represents the epoch at which the accuracy has been recorded.

4.2.2 Geographical Stratification Performance

Training was concluded after 50 epochs and we retained the model with the lowest validation loss, which occurred at epoch 15. Table 4.6 displays the final accuracy and F1-score derived for each phase on the best model. The confusion matrices for the training, validation, and testing phase are given in tables 4.7, 4.8, and 4.9 respectively. The trends visible in the confusion matrix largely follow the trend seen for the complete dataset, though deviations in accuracy and F1-measure are more prominent during this experiment. Figure 4.5 displays the loss for the model at every epoch during training, while Figure 4.6 displays the accuracy of the model at every epoch. As in the last experiment the training accuracy steadily converges to a minimum while the validation accuracy only stabilizes after 30 epochs. Remarkably, the model has its lowest validation loss recorded early on in the learning process.

Metric	Training	Validation	Testing
Accuracy	97.90%	96.59%	92.90%
Avg. Unweighted F1-score	0.9403	0.9054	0.9169
Loss	0.0771	0.1352	0.1973

Tab. 4.6: Classification performance of the best model trained on the geo-stratified dataset.

Predicted / True					F1-Measure	Top-1 Accuracy
Animals on Road	853	6	2	100	0.9099	88.76%
Flooded Road	5	1479	9	219	0.9040	86.39%
Snow on Road	4	13	3394	227	0.9594	93.29%
Negative	52	62	32	28354	0.9880	99.49%

Tab. 4.7: Training split confusion matrix (n=34,820) of the best model trained on the geographically stratified dataset. The x-axis represents the true class and the y-axis represents the predicted class.

Predicted / True					F1-Measure	Top-1 Accuracy
Animals on Road	281	2	3	41	0.8633	85.93%
Flooded Road	6	509	12	155	0.8263	74.63%
Snow on Road	4	10	1135	73	0.9478	92.65%
Negative	33	26	20	8997	0.9810	99.13%

Tab. 4.8: Validation split confusion matrix (n=11,316) of the best model trained on the geographically stratified dataset. The x-axis represents the true class and the y-axis represents the predicted class.

Predicted / True					F1-Measure	Top-1 Accuracy
Animals on Road	73	0	0	0	0.9299	100%
Flooded Road	1	54	3	0	0.9319	93.10%
Negative	10	3	48	2	0.8205	76.19%
Snow on Road	0	0	3	112	0.9782	97.39%

Tab. 4.9: Testing split confusion matrix (n=309) of the best model trained on the geographically stratified dataset. The x-axis represents the true class and the y-axis represents the predicted class.

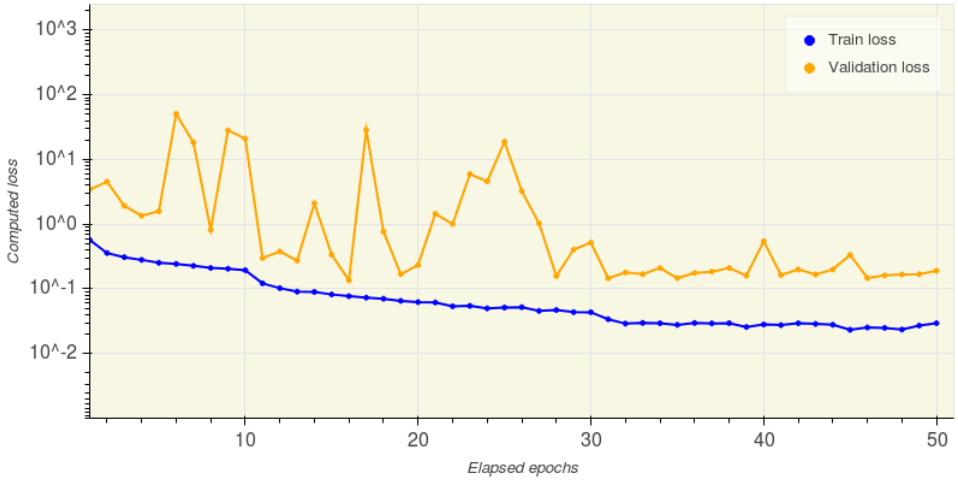


Fig. 4.5: Loss curve of the model trained on the geo-stratified dataset. The y-axis represents the loss incurred at each epoch (lower=better), while the x-axis represents the epoch at which the loss occurred. A steady decrease is indicative of a model that is converging well.

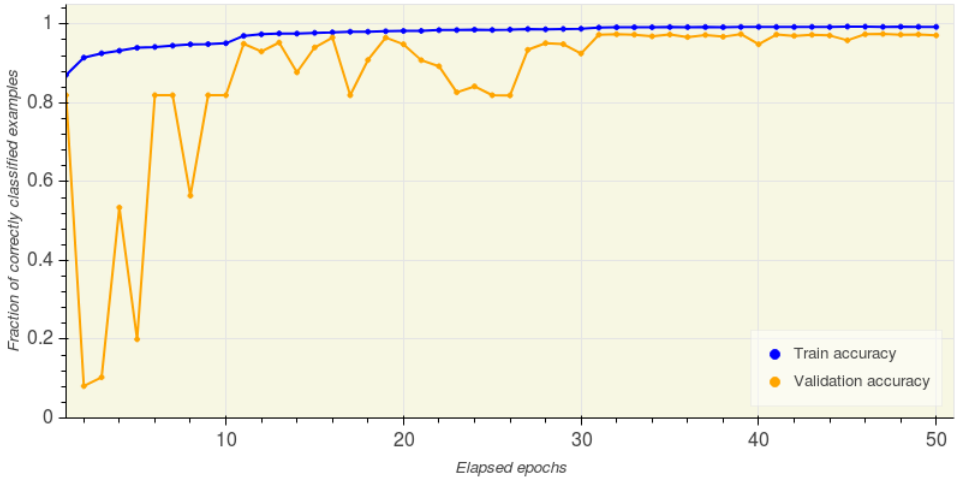


Fig. 4.6: Accuracy curve of the model trained on the geo-stratified dataset. The y-axis represents the accuracy rate at each epoch, while the x-axis represents the epoch at which the accuracy has been recorded.

5 Discussion

In this chapter we discuss the process throughout the research, as well as the results derived from it. We first consider the taxonomy and the process of creating the dataset before discussing the results derived from the model.

5.1 Taxonomy

The taxonomy proposed in this research provides a solid basis for future research that wants to consider roadscape incidents. It provides a systematic overview of incidents and accommodates sub-groupings at various levels. For instance, if a classifier is found to perform very well on a high-level grouping, then the taxonomy can be expanded with sub-groupings which can then be added as new classes. This may include expansion onto spatial characteristics as well. For instance, further research may consider the *animal on road* class in more detail while specifically aiming to classify for spatial relevance e.g. *is the animal on the road, or next to the road?*, which can be considered two separate incidents. Further, the proposed taxonomy may serve as a basis for other research efforts where spatial context and sub-grouping are important, such as illegal dumping (e.g. *rubbish on road*, or *rubbish next to the road*). For instance, it may be necessary to further sub-group different types of rubbish as well as their spatial location (e.g. *'on', 'near', or 'far from' the driving surface*) for the purposes of determining whether they are dumped or deposited in the right place.

5.2 Dataset creation

The querying strategy has proven to be effective in generating a dataset of spatially explicit images which mimic roadscape incidents. Through query construction using synonyms and breadth-wise searching we can describe a wide variety of spatial contexts while ensuring that images remain relevant to the research case. While the resulting dataset is limited in size, the classification accuracy indicates that it is effective for training purposes. The data collection and cleaning process may have progressed far quicker if the process had been outsourced, for instance using Amazon's Mechanical Turk such as in ImageNet [26]. However, this would create

uncertainty in the dataset as there may have been many interpretations of incidents. By having only one person clean the data we keep the dataset semantically consistent, which we consider crucial for the research. Since the content of the entire dataset is known and accounted for, we can infer more accurate conclusions on the dataset and ensure that the dataset does not contain false inclusions or irrelevant incidents. We can therefore guarantee that the dataset is accurate in its depiction of the target class with a uniform semantic definition. The resulting dataset created in this research is furthermore remarkable in its scope. While many large-scale image datasets used for classification focus solely on the class, images in our dataset have the explicit characteristic that they are relevant for a particular type of scenario. While it is orders of magnitude smaller than large, popular datasets, we hope that the resulting dataset may fill a niche in spatially explicit classification. With specific reference to datasets relating to on-road incidents, to the best of our knowledge, it is the first dataset that has attempts to cover roadscape unsigned physical incidents at a broad scale. It is thus a good starting point for future research in the domain of unsigned incident detection from ego-vehicles.

5.3 Incident Recognition

In this section we discuss the results and the process of incident recognition. We also discuss the t-SNE and CAM interpretation methods for the complete dataset model.

5.3.1 Model Training Process

Initially we planned to fine-tune only the fully-connected layer of the pre-trained ResNet model. In doing so, only one gradient update would have to be performed rather than for all layers, which would reduce the computation times during training. However, the resulting model vastly underperformed when compared to a fully re-trained network. We hypothesize that this is caused by the scale mismatch between typical images in the ImageNet dataset and the dataset generated in this research. Comparing Figure 3.3 (which contains example images of our dataset) to Figure 5.1 (which contains images of the ImageNet dataset) it becomes apparent that there is a scale mismatch between our dataset and the data on which the network was pre-trained. Many classes and objects in the ImageNet datasets are close-ups, whereas the examples relating to road incidents often have smaller-scale objects and areas of interest. As convolutional filters are learned from input data, the effective scale of the learned filters logically depend on the size of the object or area of interest. Hence the filters learned from the ImageNet dataset trigger on

visual cues relating to close-ups which are often not present in the incidents dataset. Re-training all layers of the network alleviated this problem.

The convergence curves of figure 4.3 and Figure 4.5 display a very erratic convergence pattern, which prompted questions on overfitting. Experiments performed during model training saw the validation loss and accuracy improve steadily if the model were to be initialized with a lower learning rate, which is reflected in the convergence curves such as in Figure 4.3. At epoch 30 the learning rate is decreased by a factor 10 to $1e^{-}5$, and afterwards there are no fluctuations as significant as earlier in the curve. However, starting out with a lower initial learning rate did not lead to better results, while the overall convergence rate would be slower. Hence, we chose to accept the erratic convergence on the validation set in favour of faster training under the assumption that the achieved local minima would not differ significantly. Based on the final accuracy and loss, this assumption appears to be correct. In part, the poor convergence during validation may be caused by the lack of dropout layers in the ResNet architecture. For every given neuron in a layer, dropout layers randomly turn off that neuron's signal by setting it to 0 with a probability of P. This excludes its signal from the classification process, and thus it trains neurons in a more balanced fashion as no single neuron becomes too influential during the learning process. Given the achieved results we do not consider this lack of dropout to be a problem.

Fig. 5.1: Example images from the ImageNet animals subset [97]. Notice how most animals are centered and prominently in view.

5.3.2 Model Classification

The high accuracy of both models is striking. It surpassed all expectations, and the implications for further research are notable. The confusion matrices show encouraging patterns, and evaluation of the F1-score confirms that the model is not overclassifying images to any particular class. While not immediately ready for deployment, the results achieved from a small dataset are encouraging for future research. Ideally, the model should have even fewer misclassifications to mitigate the amount of noise produced to the grid. However, the results were achieved using a dataset of limited size. If the accuracy is indicative of the overall task difficulty, then a larger-scale data collection effort may quickly lead to the production of a deployment-ready model. However, the high model accuracy also warrants suspicions of bias. While we took great care in limiting the influence of biases where they might occur, it is still plausible that the final result is affected by it. During data selection we attempted to emulate incidents as seen from an ego vehicle perspective, while also trying to ensure the inclusion of enough images to train a CNN. This lead to trade-offs during the selection process and the inclusion of some images which would not be included had there been enough data. Consider for instance images taken from the passenger side window. While they do capture the incident on the road itself, they might not be ideal in terms of camera parameters (pitch, yaw, angle, etc). Further research would do well to determine the degree of bias caused by the inclusion of such images. This can perhaps be combined with field-testing, e.g. through the analysis of still images from crowd-sourced driving videos of incidents. At the same time, this would give an indication of reliability during field-testing, as well as potential weaknesses of the current trained model.

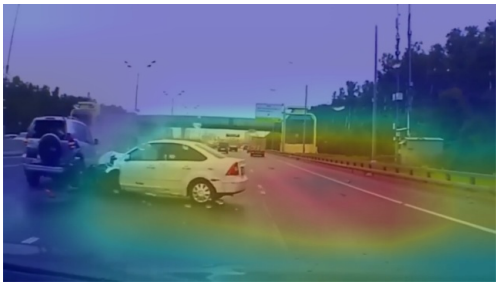
A notably consistent error of the trained model is the confusion between *Snow* or *Flooding* and the negatives class. Both classes have a very hard-to-delineate definition. Even for a human classifier it is difficult to determine whether a given cover of snow or flooding might cause problems to a vehicle passing through. This uncertainty is reflected in the consistency that misclassifications occur between the three splits. Notice also how the *Animal on Road* class is hardly misclassified during training on the full dataset, but also how it is one of the worst-off classes during validation, despite having a greater amount of training samples when compared to other problematic classes such as *Road collapse*. Inspection of misclassified images of the test set reveals that the classifier struggles with a variety of predictable problems, such as low-resolution images and blurred images. The high degree of false positives on the testing split of the model trained on the complete dataset can to some degree be explained by the observation that the classifier mistakes small and distant objects to be animals. We conjecture that the domain to classify for is broader than most other classes, as animals exist in many shapes and sizes. Further-

more, the domain is made more complex by the requirement that animals should be *spatially relevant*. That is, animals should directly be relevant to the driving situation. This is currently not trained for in the dataset, i.e. there are no animals in the negatives dataset. By adding images of animals that are not relevant for the driving situation it is possible to explicitly train for this relevance. We therefore urge that further research considers the complexity of the class domain before considering this particular class fit for field-testing.

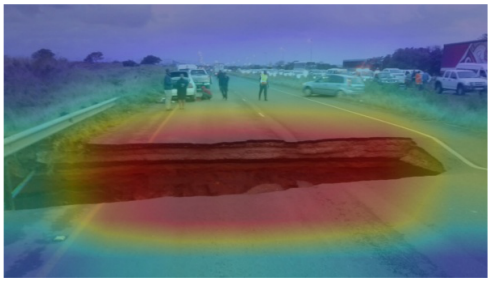
An notable observation that requires further clarification is the big swing in classification accuracy of the *road collapse* class. It goes from being one of the best classes on the training set to being the worst class during validation, but swings back to high accuracy on the test set. This can perhaps be attributed to the limited amount of samples for the class. Further research should determine whether the observed accuracy swing is a sign of poor generalization, or otherwise an outlier.

5.3.3 Classification Interpretation

In this section we discuss the model performance by interpreting visual cues derived from CAM images, as well as inspection of the t-SNE dimension reduction results. Firstly, we present the model's class attention on the prototypical images listed in 3.3. In seven out of eight cases the model correctly predicts the target class, and in six out of eight cases the model has its class attention on the right locations, as the class attention for snow is too far off the road. The CAM results are encouraging as they indicate that the model did not overfit onto irrelevant visual cues such as cliffs, at least for the given prototypical images. These results also indicate that bounding-box prediction or semantic segmentation (per-pixel prediction) is a possible avenue of research. One such option may be to perform active learning onto the CAM boundaries, where the model iteratively suggests bounding boxes or pixel regions which are corrected by a domain expert [119].

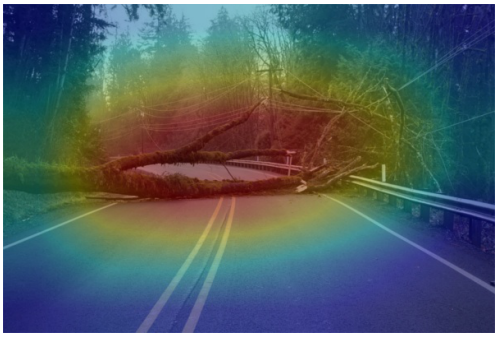


(a) Predicted: Negative [12]

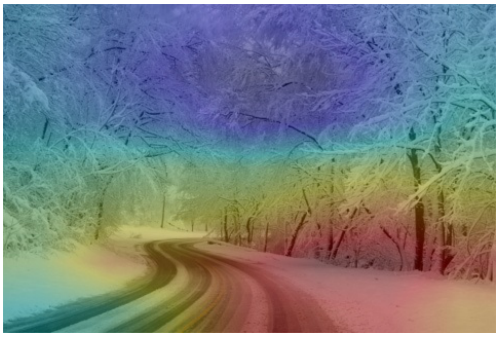


(b) Predicted: Collapse [8]

(c) Predicted: Animal [1]



(d) Predicted: Treefall [9]



(e) Predicted: Snow [95]

(f) Predicted: Flooding [79]

(g) Predicted: Landslide [105]

(h) Predicted: Fire [102]

Fig. 5.2: Class attention of predicted class overlaid on prototypical images of each class.

Interpreting t-SNE

The t-SNE plots used for inference were derived using a perplexity value of 50. With a total testing split size of 5,263 samples, the t-SNE plot considers a total of 151 nearest neighbours for each point. With a learning rate of 500 and 1,000 iterations, this configuration produces a plot which displays the underlying geometry of the classification results well. The t-SNE figure which we use for interpretation is given in 5.3. We list the other t-SNE plots in Appendix B. We have marked several remarkable clusters and outliers. It should be noted that clusters may change locations between runs because of t-SNE's random initialization of the gradient descent algorithm. As such, the relation of clusters and data-points towards one another is more relevant than their actual position on the coordinate plane. We also display the distribution of image sources after t-SNE dimensionality reduction in figure 5.4.

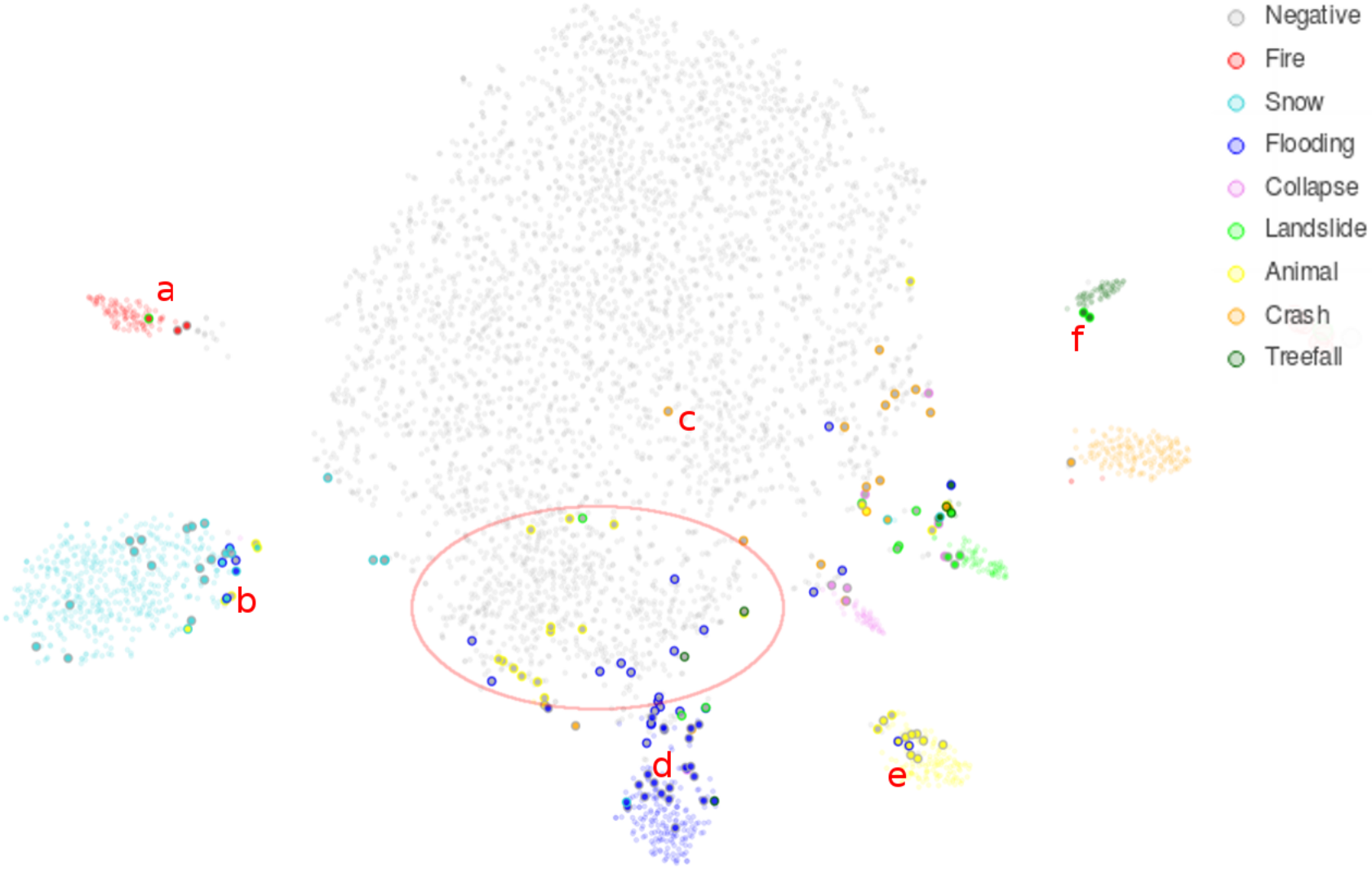


Fig. 5.3: t-SNE dimension reduction of inputs to the fully-connected layer for every image of the complete dataset model, which are used to make predictions on which class each image belongs to. Point grouping is performed with a perplexity of 50. The inner circle of each point represents its true class, with the outer circle representing its predicted class. Letters *a* through *f* represent samples which are inspected in Figure 5.5. The red ellipse indicates a region consisting of almost exclusively of geograph negatives.

Several patterns appear in the t-SNE plots:

- **The model can tell most classes apart with great ease**
  Most positives classes are clustered together without a fuzzy border towards the grey negatives cluster, with the exception of the *flooding* and the *landslide* classes. The uncertainty of these classes is reflected in the reported accuracies for these classes, while the other classes are far less affected by uncertainty throughout all three splits.

- **The *flooding* class has the greatest uncertainty in its classification region**
  As indicated by point of interest *d*, the *flooding* class strongly gravitates towards the Geograph negatives cluster and shares a large indecisive boundary region with it. This overlap is consistent between t-SNE parameter settings and can also be seen in the other t-SNE plots in appendix B.

- **Images within the negatives set are easily distinguishable within the negatives cluster**
  Outlined with a red ellipse we find a cluster that predominantly consists of negative Geograph images. We conjecture that this cluster is formed by the comparatively greater amount of countryside images within the Geograph negatives set when compared to both the Berkeley Deep Drive (BDD) and the Cityscapes negatives. We explored this observation further in figure 5.4, which in turn reveals that images within the negatives cluster are notably dissimilar, which the model is able to pick up. At a glance, the BDD subset provides the greatest spread within the negatives cluster, which indicates that it provides the most diverse data for training purposes. Notice too how there is a visible separation between the BDD cluster and the Geograph cluster, despite the fact that they visibly belong to the same macro cluster. This is a strong indication that the model perceives them as dissimilar.

- **There are no clusters segregated by source within the clusters of positive classes**
  As seen in figure 5.4, positives clusters containing Geograph images do not contain distinguishable sub-clusters. While the model has the tendency to see each source as similar, they are not so similar that they're considered to be separate. This is an encouraging observation as it indicates that the potential for bias as a result of source dissimilarity is likely limited.

  In figure 5.5 we give an overview of images from the highlighted areas of interest of figure 5.3.

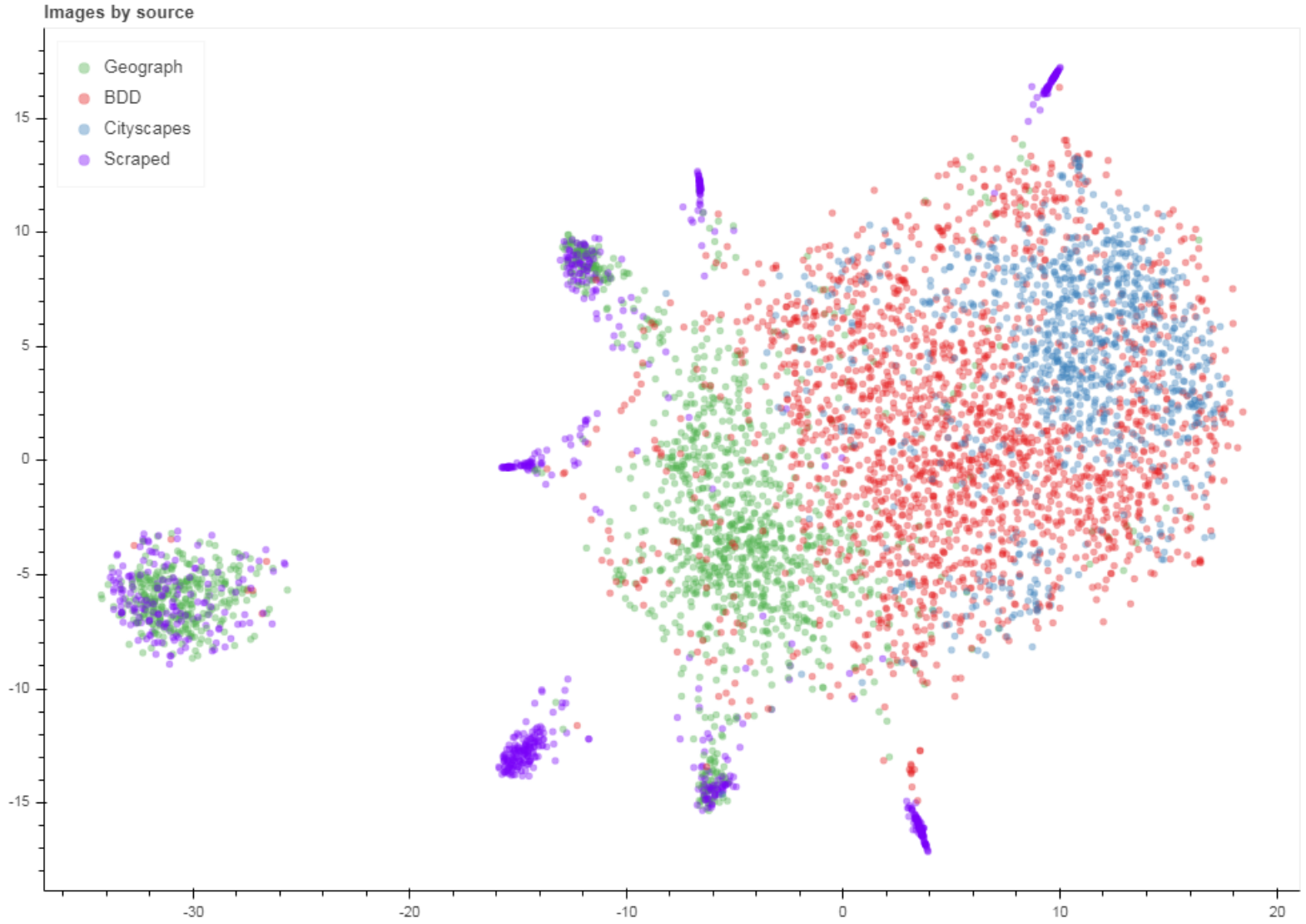


Fig. 5.4: Distribution of image data sources after t-SNE clustering with a perplexity of 50. Note that this t-SNE plot was computed with a different initialization but with the same parameters as figure 5.3, and thus reflects the same global patterns.

Interpreting misclassified CAM Images

(a) *Landslide* misclassified as *Fire* (uncoloured)

(b) *Animal on Road* misclassified as *Negative*

(c) *Crash* misclassified as *Negative*

(d) *Flooding* misclassified as *Negative*

(e) *Negative* misclassified as *Flooding*

(f) *Landslide* misclassified as *Treefall*

Fig. 5.5: Outliers identified during the t-SNE dimension reduction process. Where applicable, we overlay the class attention for the predicted class on a violet (low attention) to red (high attention) color ramp.

for each example listed in 5.5 we discuss observations from the images.

- **5.3a**: We display this image without overlaying class attention as the cause of the misclassification is quite clear. The image has a high degree of yellow-red tones visible, and the boulder on the road is not clearly situated on the driving surface.

- **5.3b**: This particular case is least explainable of all. Its spatial position would indicate that it should be classified as flooding, while it is misclassified as a negative. There are no clear clues as to why this happens. The class attention model does not provide an insightful explanation either. Most remarkable is that the model considers the image similar to snow, yet still classifies it as a negative. We remain inconclusive on what is happening during this classification.

- **5.3c**: The low resolution of the original image may make it hard for the model to determine that a car is rolling over. Furthermore, such representations are rare in the dataset, so it may be case of not having seen enough representations to correctly classify it.

- **5.3d**: Here we see the semantic uncertainty of the *Flooding* class. The puddle on the right looks significant enough to be considered flooded, which is why it was included as a positive. However, the model classifies it as a negative, perhaps due to the lateral offset to the road and the lack of distinctive features on the puddle. Many such misclassification in this general cluster share this characteristic.

- **5.3e**: The object highlighted by the class attention is in fact a car parked on the side of the road. We hypothesize that the hard-to-see details at this distance makes it likely that it is mistaken for an animal. Furthermore, the flooding is far away in the image as well, which would decrease the likelihood of a correct classification further.

- **5.3f**: The misclassification to fallen trees is perhaps caused by the leafy material in the landslide debris. There are several such images in the dataset, but this appears to be the only misclassified one.

5.3.4 Geographical Stratification

In driving scenarios it is imperative that trained models generalize well across various regions or landscapes. A model needs to detect incidents regardless of the terrain that the vehicle is in. This is especially the case for classifiers tasked with recognizing hazardous situations. For instance, if an incident detector is unable to recognize a flooding in a desert (e.g. during a flash-flood) it may result in an autonomous vehicle that drives straight into a hazardous situation. Being able to generalize well to unseen geographical regions is therefore an important consideration for a model's fitness to deploy and quality. Although Wales as a region is quite similar to the rest of England and Ireland, the accuracies and F1-measures achieved on the test set are indicative that the trained model remains stable in unseen regions with a similar geographical layout as the trained model achieves high accuracy values for the applicable classes in a geographical region which it has not been trained in. This is encouraging for the eventual deployment of models trained to recognize incidents, as incidents should be learned across the entire domain. The trained model's performance is furthermore indicative that geographical correlation within the dataset can be largely ruled out, further solidifying the hypothesis that the trained classifiers are reliable. The next step in determining the capacity of incident classifiers to generalize should be a new experiment that contains data from a very dissimilar landscape, such as Asia.

Remarkable observations from the confusion matrices are the varying degrees of accuracy for the *flooding* class across the three splits, and the lower accuracy of the *negatives* class during testing. During validation, the *flooding* class is misclassified much more often than during training and testing. This is perhaps caused by the limited sample size, or possibly due to a poor model balance resulting from the best model being recorded early in the process as we used *loss* as an indicator of model quality. The best recorded model also occurred early in the training process while the model was still converging, which may have resulted in an imbalanced decision process. A larger test sample size may result in more conclusive results on model balance.

5.4 Limitations

5.4.1 Limitations of the Taxonomy of Incidents

The taxonomy is currently limited in its differentiation between man-made and natural objects. While incidents such as fires relating to road networks are often a man-made phenomenon (e.g. a car burning after a wreck), natural fires may be a likely cause of disruptions as well. We observed that the taxonomy would quickly become an intensely discussed topic during presentations, and meetings. While it was remarked that it is not necessarily incorrect, it was noted that it would be hard to determine which attribute each group belongs to. Fire can both have a man-made and a natural cause, and determining what caused it belongs to may be difficult. The current taxonomy is governed by a *most likely cause* distinction, which is an ill fit for a full consideration. One potential solution may be to consider such incidents in both groupings with a stringent definition to differentiate between both. However, the question then remains whether such closely-related incidents can be recognized from images, and whether it is necessary to be able to recognize them as distinct incidents. While the source of the incident reflects a reality about each target class, it is perhaps impossible to create a ruling that accurately captures this uncertainty from image-based sensors.

An addition to the process of creating deeper groupings in the taxonomy is to consider the concept of synsets (sets of related synonyms) such as those used in ImageNet [26]. Princeton's WordNet [68] may form a good basis for some of the classes in this research which are not combinations of various terms. For instance, *landslide* is listed as a distinct synset, along with its hypernyms *rockslide* and *mudslide*, while *animal on road* is not listed as it is a combination of *animal* and *road*. Instead, the various hypernyms of *animals* may be considered separately, and then combined with their context term (e.g. *road*). For classes which have hypernyms, standardized search terms should be considered as much as possible so that the semantic definition will remain the same throughout future research efforts.

5.4.2 Dataset Creation Limitations

Images gathered by API harvesting contained many duplicates prior to the selection of images. We filtered many of these duplicates by checking each image with every other image for their exact equivalence without considering resizing and artifacts. However, this filtration process is imperfect as we only check for exact equivalence between images. This means that resampled, resized, and images which have a filter applied to them are retained in the dataset. While we suspect that few duplicate

images have been saved during the cleaning process, it is worthwhile to consider cleaning any remaining duplicates out further. A better solution may be to use feature extraction methods, such as perceptual image hashing using feature points which is able to accurately detect equivalence while accounting for a wide variety of distortions, transformations, and alterations [71].

The data used in this project relies on single labels for classification. This limits the model to only being able to detect a single incident at a time. We did not consider multilabel-classification to be immediately relevant for the case as multiple incidents occurring at once is unusual, and thus it would be hard to build up a sufficiently diverse dataset. Multilabel-classification would require a different model set-up as well, for instance by training an ensemble of binary classifiers (a true/false classifier, one for each class to indicate whether it is present) at the end of the model for each output class instead of a single classifier.

5.4.3 Incident Detection Limitations

Due to the limited amount of images available for most classes the model is expected to be limited in the amount of representations that it can recognize. Classes with a difficult semantic definition such as flooding (e.g. *when is the road covered enough to be considered flooded?*) may especially be affected by this limitation. A stringent definition of semantically unclear classes may help to build a dataset that has a less uncertain decision boundary. The lack of dynamic information in static images may also make the detection of certain classes more difficult. Based on the confusion matrices generated, detecting flooding from static images has been observed to be a more difficult case than a well-delineated incident such as a landslide. The inclusion of information from continuous (video) data may help to improve the classification accuracy for the flooding class, as being able to see the turbulence and waves on the water may aid in determining that water is covering the surface. Data fusion methods such as using LIDAR data may help to further differentiate a normal driving surface from a covered one, as the return of the laser beams originating from the device will be different depending on the cover (e.g. loss of signal for a wet surface).

The model may furthermore be limited by its spatial interpretation of certain classes. Ideally, incidents in question should only be considered an incident if they are relevant to the road itself. All images gathered during this research have this feature, either explicit (e.g. *cow is standing on the road*) or implicit (*sheep standing on a hill next to the road without a fence*). However, in many cases there has not been a robust set of negatives tailored specifically to differentiate between incidents that are immediately relevant to the road network and those that are not. There may thus

be some uncertainty on whether the model is able to tell apart relevant incidents from non-incidents. In this research we inspected the possibility for the class *snow* to become spatially relevant. Based on the classification results it appears that this approach may be successful. A visual confirmation for one prototypical case is given in figure 5.6, where the model successfully has its highest class attention based on the snow on the driving surface rather than snow elsewhere in the image. Further research should run experiments to test to which degree it is able to determine relevance for other classes.

Lastly, there is a small chance that the full dataset classification may be sensitive to biases as a result of the way that the negatives dataset was generated. The negatives set of the full experiment does not contain scraped images which differ from the Geograph negatives and the driving datasets. There is thus a chance that this model has learned to distinguish scraped positive from non-scraped negatives. The geographically stratified model does not suffer from this suspected bias in that it is well-formed. That is, the test dataset contains the same sources as the training and validation data, and scraped images are only used to enhance the model during the training and validation process. Here, we still see the same high performance on the test set, which indicates that the task can likely be solved well even when accounting for possible biases. We thus consider the bias of choosing different sources between the training and testing dataset to be a remote possibility. While there are too few misclassifications to draw conclusive numbers on potential biases towards each source at this moment, Figure 5.4 supports the hypothesis that any potential biases by sources are a minor influence at worst, as there is no clear separation of scraped and Geograph images within the three clusters of positive classes. For a decisive test on the influence of biases by source type we suggest that a new experiment is run with a second curated test dataset that also contains scraped samples so that the degree of bias can be assessed.

5.4.4 Limitations of the Geographical Stratification

From the test accuracy of the geographical stratification we can infer that learned incidents can be recognized in new regions. Furthermore, we can rule out that geographical correlation has played a decisive role in the high accuracy. While the resulting model is less accurate than the main experiment, the reported accuracy still greatly exceeds our expectations. While this means that we can largely rule out immediate geographical correlation, the difference in landscape between rural England, Ireland, and Scotland does not differ much from Wales, so it is not known how well the model generalizes to a different country or continent altogether.

Fig. 5.6: An image of rural Iceland with snow on the driving surface. We overlaid the class attention for the class *snow* on a blue to red color ramp. Red colors indicate a higher class attention. The model's confidence for the class snow in this image is 99.5%. Notice how despite that the image contains snow everywhere, the only snow in focus is on the surface itself. Original image source: [47]

6 Conclusion

The road network is under increasing pressure as car ownership rises and road transport intensifies. This increase in road network pressure intensifies the effect that incidents have on the road network. At the same time, vehicles equipped with sensors are becoming increasingly prevalent on the road network as autonomous vehicles are beginning production. To our best knowledge, no existing research has previously been performed on the recognition of incidents as a domain using images as seen in sensor-equipped vehicles. This formed the motivation for this research, namely to assess incident detection as a domain. In this research we have created a taxonomy for unsigned physical incidents, gathered a dataset of images to be used in classification, and investigated whether unsigned physical incidents are learnable by convolutional neural networks. Based on the results and the discussion we draw up the following conclusions to our research questions:

- **RQ1: How can incidents be assigned to a typology for the purposes of classification?**
  Incidents can be assigned to a typology by considering their commonly shared attributes through a taxonomy structure. By assigning groupings as part of a taxonomy incidents can be evaluated by similarity, which limits the amount of classes needed during classification. This lets incidents be classified with varying levels of depth, such as *animal on road* as a general concept versus specific species of animals. In doing so, the taxonomy can be expanded and adjusted to fit local conditions while retaining a common hypernym between taxonomies. The taxonomy workflow is not limited to this research specifically. The taxonomy workflow can serve as a basis for research where attribute-specific groupings have to be assigned and synonym trees such as Princeton's WordNet cannot be considered.

- **RQ2: How do we create an image dataset of unsigned physical incidents?**
  A dataset can be created by querying search engine APIs and image hosting initiatives for images of incidents using crafted queries by considering synonyms, as well as translating queries into other languages. By searching breadth-wise using synonym pairs, semantic concepts such as *snow* can be expanded into related terms like *blizzard* and combined with context terms such as *street*, which denotes an urban context, to harvest images from a diverse

set of representations. The developed methodology enables sampling efforts for datasets that require the inclusion of context terms to gather thousands of images which span the context domain. The dataset generated during the research has the intrinsic trait that every recorded image of unsigned physical incidents is *spatially relevant* to the road network. This enables opportunities for testing how classifiers react to learning spatial relevance within classes. Lastly, the created dataset is a first of its kind in that it is spans the domain of incidents beyond one individual class. We hope that this dataset serves as a basis for further research into unsigned physical incidents.

- **RQ3: How accurately can convolutional neural networks detect unsigned physical incidents using an image dataset?**
  A Convolutional neural network using the ResNet-34 architecture is able to detect our subset of incidents with an overall accuracy rate of accuracy rate of 97.15% and an average unweighted F1-score of 0.8909. We can thus conclude that CNNs can very accurately classify unsigned physical incidents using an image dataset. The achieved accuracy far exceeds initial expectations, and it is a strong indicator that unsigned physical incidents as a domain can be learned well using CNNs. It should be noted that the final results may suffer from bias due to an imbalance in the type of images between the negatives and positives dataset, though it is not expected to be a major factor in the classification accuracy.

- **RQ4: How stable is the classification of unsigned physical incidents?**
  In this experiment we have trained a second model on three classes (*snow*, *flooding*, and *animal on road*) for which we have geo-tagged images. By stratifying three types of incidents to England, Scotland, and Ireland for training/validation and by testing on the region of Wales, we have proven that incident recognition is stable between similar geographical regions. Furthermore, we can rule out immediate geographical correlation as a result of sampling, reinforcing the hypothesis that incidents as a domain can be learned well. The overall accuracy of this second experiment was an accuracy of 92.90% with an average unweighted F1-score of 0.9169. The achieved accuracy indicates that unsigned physical incidents as a domain can be generalized well, which is an important consideration for operational models as they must be able to function in any environment and under any circumstances.

Based on the answers to the research questions we can thus prove our hypothesis and answer the main research question as follows:

> **Main Question:** *How can we automatically detect unsigned physical incidents from sensors that can be mounted on driving vehicles?*
>
> To answer our main question we have first created a classification system by setting up a taxonomy for unsigned physical incidents, and subsequently we have web-harvested a dataset. After training two CNNs models to determine how well unsigned physical incidents may be learned by a classifier we draw up the following conclusion:
>
> **Conclusion:** *If the assumption that the training procedure has been minimally influenced by biases holds, then CNNs can accurately classify unsigned physical incidents in RGB images derived from cameras mounted on driving vehicles by web-gathering images of incidents that are representative of driving scenarios. Furthermore, classification was found to be well-generalizable between similar environments.*

To summarize, in this research we have created a taxonomy of unsigned physical incidents, a classification dataset of unsigned physical incidents, and two pre-trained models capable of recognizing unsigned physical incidents. Through the high accuracy achieved during the research we have proven that unsigned physical incidents can be recognized well, even in unseen but similar geographical areas. We hope that this research has formed a basis for further research in incident detection from ego-vehicles as a domain, which was previously under-researched, as our results indicate that it may be made operational with due diligence.

7 Recommendations

In this chapter we give our recommendations for future research. Firstly, we consider the taxonomy and the dataset creation process before we suggest improvements to the model. Lastly, we reflect on general insights that may be taken away from the research.

- **Future research should consider the taxonomy of incidents in more depth**
  In this research we briefly considered how incidents may be formalized from groupings of attributes. Future research should consider in more depth which incidents and attributes can be used to deepen the taxonomy. Further research may also consider how incident groupings can use make use of Synsets as in [26] to generate formalized semantic groupings so that common hypernyms such as *natural* or *man-made* may stay constant in definition.

- **Further data gathering efforts should consider the use of outsourcing services such as Amazon's Mechanical Turk**
  In this research we considered the integrity of the dataset integral to determining whether incidents could accurately be detected from incidents. Now that this hypothesis has been proven it may be worthwhile to consider upscaling efforts and to determine whether mass-gathered images with a fitting cleaning regime may help to create a richer dataset and to process images faster than may be achieved by manual cleaning. The trade-off is that the rigid semantic definition that has currently will expand, which future research may have to account for.

- **Test the potential bias of assuming that scraped images are equal to images from driving datasets**
  In the full dataset experiment we did not include scraped images in the negatives dataset. This may be a potential source of bias as the model may have learned to distinguish the scraped images from the other sources. While we do not consider this to be a major risk as the geographically stratified experiment indicates that a high accuracy can be reached regardless, further research would do well to assess the degree of bias caused by this decision.

The high accuracy of the resulting models are encouraging indicators that the task can be learned well. However, several issues prevent us from suggesting that it is ready for deployment. Firstly, while the model accuracy is impressive, there are still too many false positives and false negatives. The implication is that too often the model would send out a false signal. If it were to be broadcasted, it might close a road while there is no reason for it. The model should therefore have even fewer false positives before deployment should be considered. We give the following recommendations to improve the current model to a deployable state.

- **Further research should test the classification accuracy on lightweight models**
  In this research we opted to use a deep CNN to determine the degree to which incidents may be learned from images. Given that many features have to be detected for an AV to drive autonomously, it is desirable to use lightweight model for tasks that are less crucial. Lightweight models require fewer parameters and operations during runtime, which results in faster processing speeds, and thus for frames to be processed faster. Given the achieved accuracy we propose that further research considers the use of lightweight models for this task to test whether the task can be offloaded to a model that is less resource-intensive. Figure 2.11 gives an overview of such models. Models with a parameter count and number of operations similar to ENet can be considered lightweight.

- **Consider field-testing the model to determine how it will behave in practice**
  The current model has been trained on a sparse dataset, which means that it has been trained on a limited set of the many possible representations of the incident classes. It is therefore plausible that the model may not perform well in field-testing. We suggest to perform field-testing with the model on video sequence data to determine how the model will respond to actual driving scenarios rather than web-gathered images. For instance, this can be done using videos recorded from a dashboard camera near incidents, which may then be fed to the model to determine whether it can pick up the incidents in each video.

- **Experiment with localization and spatial relatedness**
  In this research we have touched upon the concept of spatial relatedness. Specifically, we do so by assessing whether the classifier can correctly determine that images that contain snow along the road but not on the road itself is not in fact an incident. However, this process may be made explicit. As an example for the snow class, for a given image the driving surface may first be predicted, and subsequently a classifier may check pixel-wise for snow in that image. the classifier can then predict the amount of snow onto the driving surface by masking the image to only the driving surface. This may help to differentiate edge cases and to create a hard definition of how much snow is accepted for driving conditions. Similar cases may be devised for other incidents, such as animals in close proximity to the road. Research such as in [38] may be considered for this case.

Lastly, we propose that this research may serve as an example for other training efforts where a clear semantic definition and dataset are lacking. For instance, consider the problem of illegal dumping. Debris, garbage bags, and other trash are not a problem in itself so long as they are in the correct location. However, a pile of garbage along a country road with no clear indication of ownership strongly indicates that it has been dumped. Such a research set-up would benefit from following a similar approach, where normal conditions can be discriminated from anomalous situations within a particular context.

Appendix A: Data Retrieval

This appendix contains auxiliary information about the data retrieval methods used in the research.

Expanded Queries

This section contains the expanded queries used when harvesting web sources.

We create our queries by combining search terms from two subsets. The first subset of *context terms* accounts for the context in which incidents may occur. The second subset covers the representations of the incident in question. For instance, in the combination set {*'snow on', 'blizzard on'*} and {*'street', 'road'*} where representations for snow are the representation and the street types are context terms, there exists four combinations; *'snow on street', 'blizzard on street', 'snow on road'*, and *'blizzard on road'*. Depending on the incident in question we may use context terms either as prefixes or suffixes, whichever approximates natural language the closest.

Context Terms

Driving Surfaces
We used the terms {*'road','highway','street'}, and 'route'*} as context terms for driving surfaces. We chose these terms to represent various environments. Our reasoning is that roads and streets are general place indicators found in urban areas, while highways and routes are more prevalent in non-urban areas.

Vehicles
The terms *'lorry', 'truck', 'vehicle','motorcycle','highway'*, and *'dashcam'* were used to denote types relating to vehicles. While implicitly not vehicles, we include the latter two terms as they often relate strongly to incidents. For instance, a *car crash happening on a highway* may simply be shortened to a *highway accident*. Likewise, a

Code	Incident	Context	Query Terms	Queries
MAN-O-1	Crash	Vehicles pre-fix	{*'crash'*, *'accident'*, *'collision'*}	18
MAN-C-1	Collapse	Surfaces pre-fix	{*'collapse'*, *'sinkhole'*, *'destroyed'*}	12
MAN-C-2	Fire	Vehicles pre-fix	{*'burning'*, *'on fire'*}	12
NAT-O-1	Animal on road	Surfaces suffix	{*'animal crossing'*, *'animal standing on'*, *'deer on'*, *'sheep on'*}	16
NAT-O-2	Treefall	Surfaces suffix	{*'tree blocking'*, *'fallen tree on'*, *'treefall on'*}	12
NAT-C-1	Snow on road	Surfaces suffix	{*'snow'*, *'blizzard'* , *'snow city'*}	12
NAT-C-2	Flooding	Surfaces suffix	{*'flooding on'*, *'overflowed'*, *'submerged'*}	12
NAT-C-3	Landslide	Surfaces suffix	{*'landslide on'*, *'mudslide on'*, *'rockslide on'*, *'boulder on'*, *'rocks on'*, *'rockfall on'*}	24

Tab. 7.1: Queries performed to retrieve data

car crash recorded on a dashcam may be shortened as a *'dashcam crash'*. Depending on the incident in question we apply them either as prefix or as suffix. We denote this for each incident individually.

Multilingual Queries

The following terms were used to query for multilingual queries:

	A	B	C	D	E	F	G	H	I
1		**Animal on Road**	**Collapse**	**Fire**	**Flooding**	**Landslide**	**Snow**	**Treefall**	**Vehicle Crash**
2	Dutch	Koe op weg	Wegverzakking, zinkgat weg	Brandende auto	Overstroomde weg, Wateroverlast straat/weg	Aardverschuiving over weg	Sneeuw op straat, sneeuwoverlast op straat	Omgewaaide boom, omgewaaide boom op weg, stormschade boom weg	auto ongeval, verkeersongeval
3	Croatian	životinja na cesti		požar na autoputu		klizište na cesti, klizište prekrilo cestu	snijeg na cesti	stablo na cesti	prometna nesreća
4	Farsi		فرورفتگی جاده		آبگرفتگی معابر		جاده برفی	افتادن درخت در جاده	تصادفات رانندگی
5	Mandarin	马路动物	公路塌陷, 马路塌陷	公路火灾	马路淹水, 道路淹水, 公路淹水	公路 塌方, 道路 塌, 道路 塌方	马路积雪		交通事故, 撞车
6	Slovak			poziar cez cestu	zaplavena vozovka/cesta /ulica	zosuv pody na ceste	ujazdeny sneh na ceste/vozovke, sneh na ceste, zasnezena cesta , cerstvy sneh na ceste/ulici/vozovke/	strom cez cestu	auto nehoda/havaria, dopravna nehoda

Fig. 7.1: Queries performed per language for each class.

Appendix B: t-SNE plots

In this appendix we list the t-SNE plots that were produced using a perplexity of 20, 100, 200, and 250. Note that the spatial structure differs between runs because of the random initialization of the gradient descent algorithm of t-SNE.

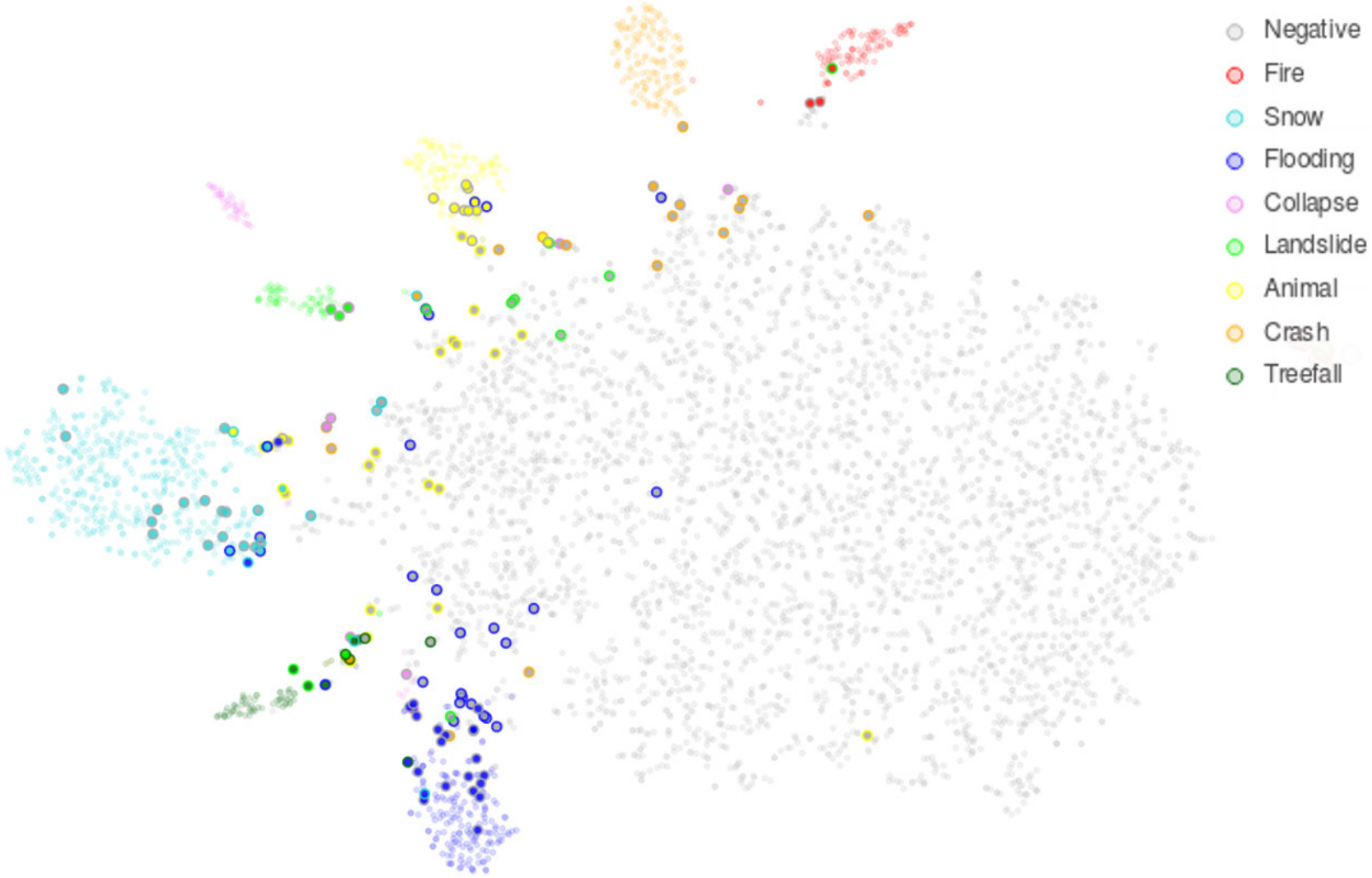


Fig. 7.2: t-SNE dimension reduction of inputs to the fully-connected layer for every image of the complete dataset model, which are used to make predictions on which class each image belongs to. Point grouping is performed with a perplexity of 20. The inner circle of each point represents its true class, with the outer circle representing its predicted class.

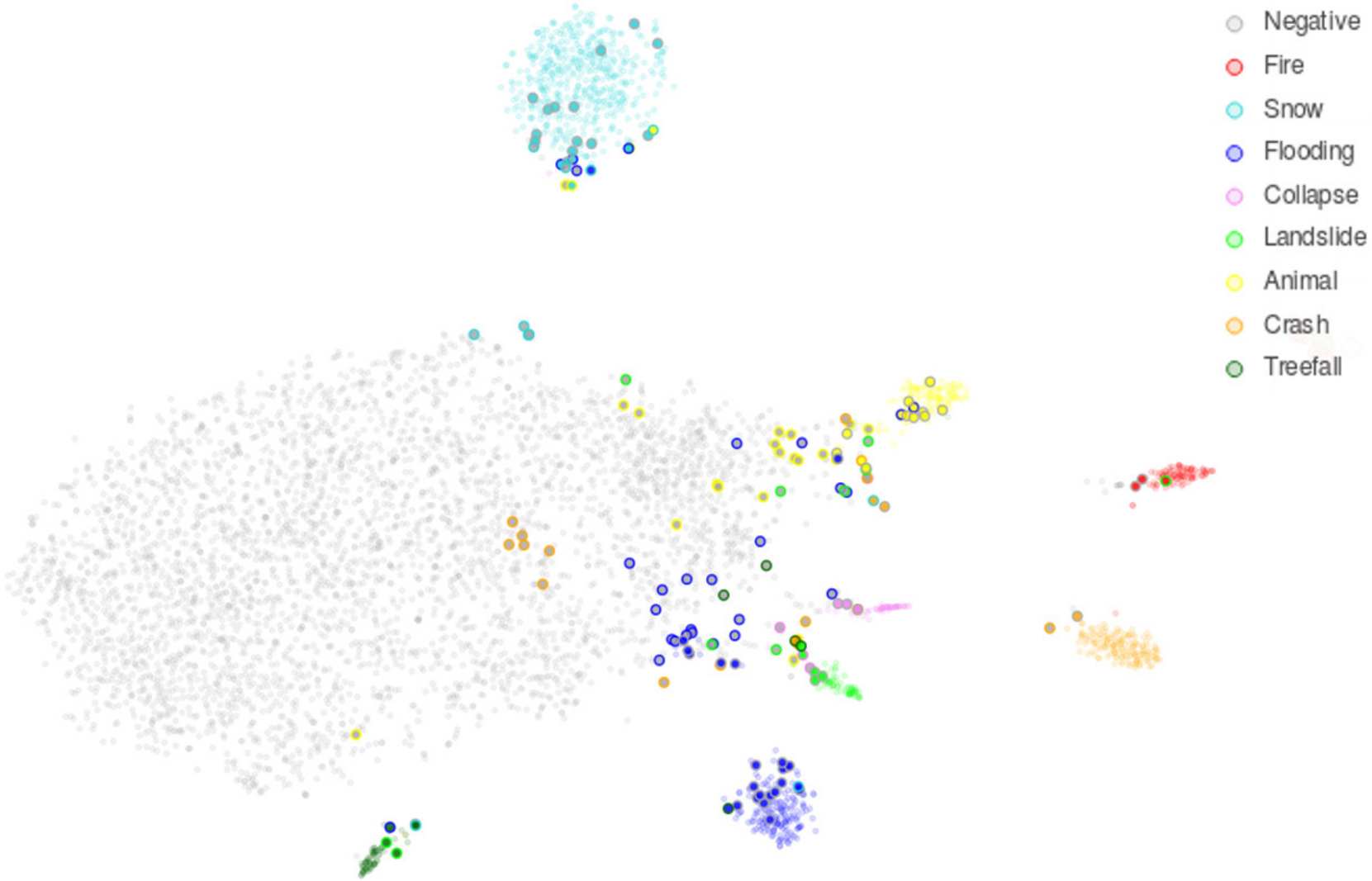


Fig. 7.3: t-SNE dimension reduction of inputs to the fully-connected layer for every image of the complete dataset model, which are used to make predictions on which class each image belongs to. Point grouping is performed with a perplexity of 100. The inner circle of each point represents its true class, with the outer circle representing its predicted class.

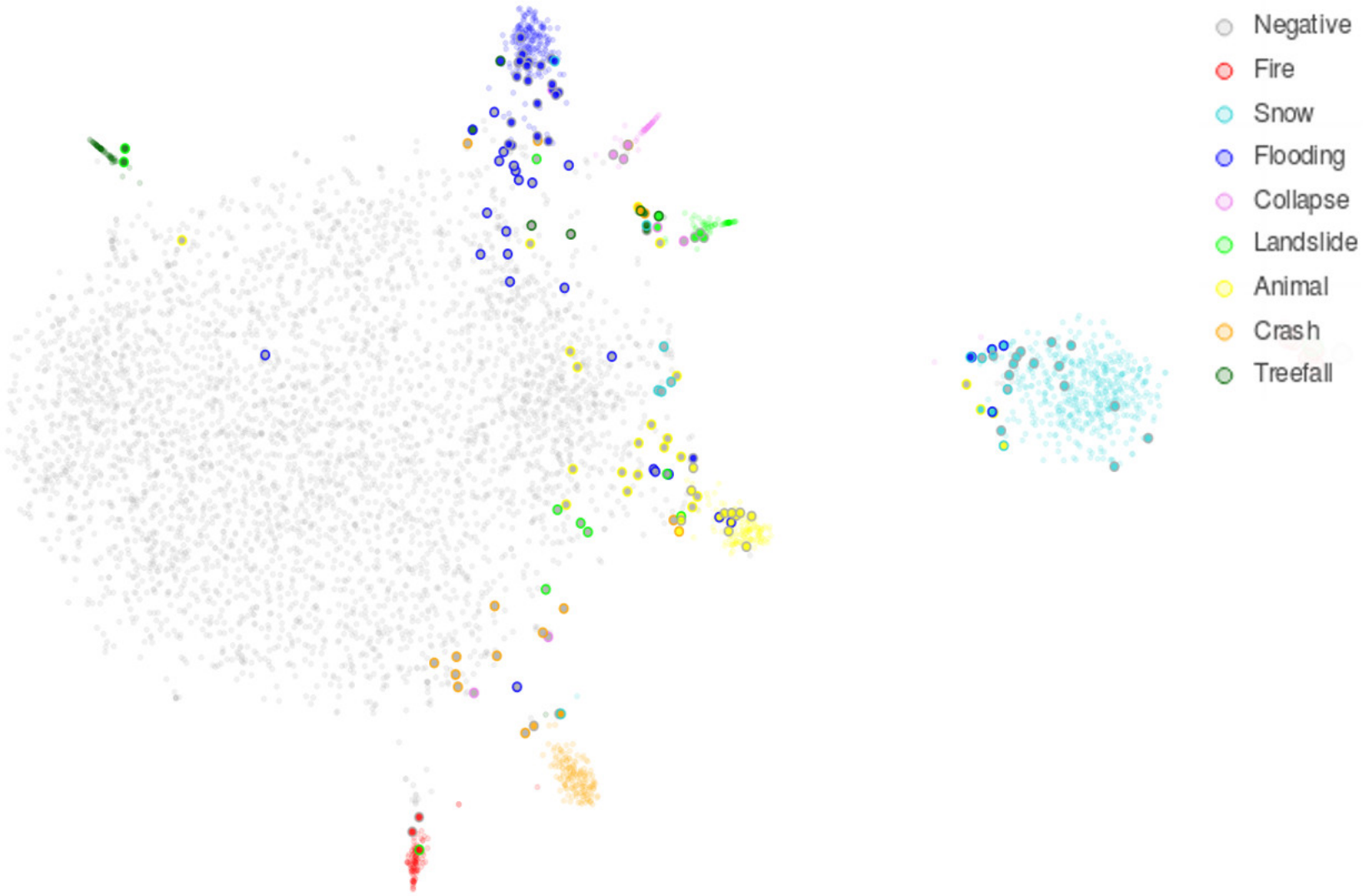


Fig. 7.4: t-SNE dimension reduction of inputs to the fully-connected layer for every image of the complete dataset model, which are used to make predictions on which class each image belongs to. Point grouping is performed with a perplexity of 200. The inner circle of each point represents its true class, with the outer circle representing its predicted class.

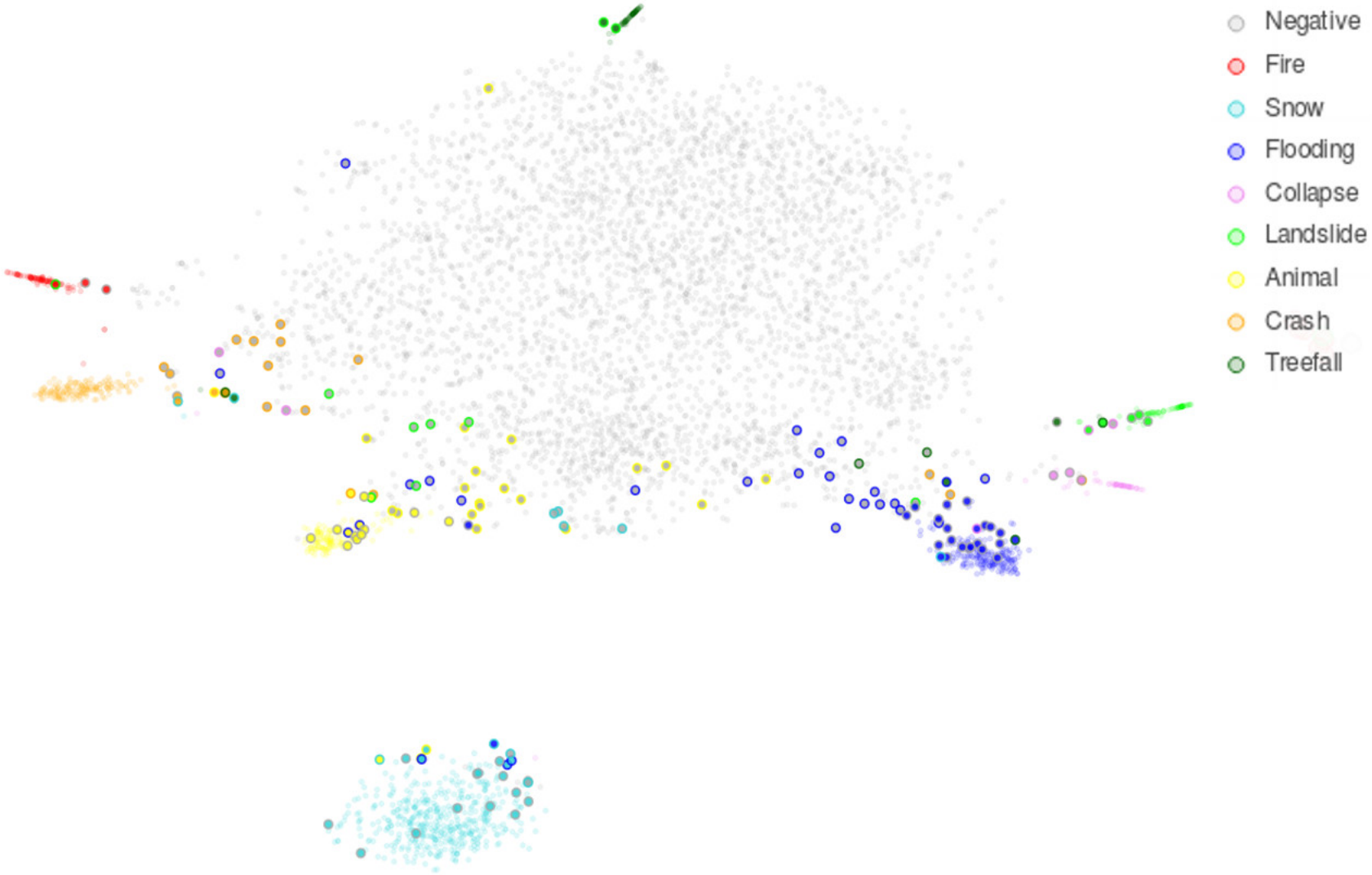


Fig. 7.5: t-SNE dimension reduction of inputs to the fully-connected layer for every image of the complete dataset model, which are used to make predictions on which class each image belongs to. Point grouping is performed with a perplexity of 250. The inner circle of each point represents its predicted class, with the outer circle representing its predicted class.

Bibliography

[3] Charu C. Aggarwal. *Data Classification: Algorithms and Applications*. Google-Books-ID: nwQZCwAAQBAJ. CRC Press, Sept. 15, 2015. 710 pp. (cit. on p. 12).

[4] Chris Aldrich and Lidia Auret. *Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods*. Advances in Computer Vision and Pattern Recognition. London: Springer-Verlag, 2013 (cit. on p. 14).

[5] Oleg Alexandrov. *An illustration of the gradient descent method. I graphed this with Matlab*. Version 654. Nov. 19, 2004 (cit. on p. 16).

[7] A. Anwar, T. Nagel, and C. Ratti. „Traffic Origins: A Simple Visualization Technique to Support Traffic Incident Analysis“. In: 2014 IEEE Pacific Visualization Symposium. Mar. 2014, pp. 316–319 (cit. on p. 7).

[11] Katja Berdica. „An introduction to road vulnerability: what has been done, is done and should be done“. Version 359. In: *Transport Policy* 9.2 (Apr. 1, 2002), pp. 117–127 (cit. on p. 5).

[12] Best Test. *Crazy highway car crash - dashcam*. Version 581 (cit. on pp. 36, 64).

[13] Xin Bi, Bin Tan, Zhijun Xu, and Libo Huang. „A New Method of Target Detection Based on Autonomous Radar and Camera Data Fusion“. In: Sept. 23, 2017 (cit. on p. 9).

[14] Jack Boeglin. „The Costs of Self-Driving Cars: Reconciling Freedom and Privacy with Tort Liability in Autonomous Vehicle Regulation“. Version 45. In: *Yale Journal of Law and Technology* 17 (2015), p. 171 (cit. on p. 2).

[15] Léon Bottou. „Large-Scale Machine Learning with Stochastic Gradient Descent“. In: ed. by Yves Lechevallier and Gilbert Saporta. Physica-Verlag HD, 2010, pp. 177–186 (cit. on p. 17).

[16] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. „Learning multi-label scene classification“. Version 515. In: *Pattern Recognition* 37.9 (Sept. 1, 2004), pp. 1757–1771 (cit. on p. 12).

[17] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. G. Jung. „A New Approach to Urban Pedestrian Detection for Automatic Braking“. Version 752. In: *IEEE Transactions on Intelligent Transportation Systems* 10.4 (Dec. 2009), pp. 594–605 (cit. on p. 9).

[18] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. „An Analysis of Deep Neural Network Models for Practical Applications“. Version 900. In: (May 24, 2016) (cit. on p. 24).

[19]M. Augustine Cauchy. „Methode generale pour la resolution des systemes dequations simultanees“. Version 652. In: *Comptes Rendus Hebd. Seances Acad. Sci.* 25 (Oct. 1, 1847), pp. 536–538 (cit. on p. 15).

[20]L. Chen, J. Yang, and H. Kong. „Lidar-histogram for fast road and obstacle detection“. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). May 2017, pp. 1343–1348 (cit. on p. 9).

[21]Anna Choromanska, Yann LeCun, and Gérard Ben Arous. „Open Problem: The landscape of the loss surfaces of multilayer networks“. In: Conference on Learning Theory. June 26, 2015, pp. 1756–1760 (cit. on p. 15).

[22]Dan Claudiu Cirean, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. „Deep, Big, Simple Neural Nets for Handwritten Digit Recognition“. Version 903. In: *Neural Computation* 22.12 (Sept. 21, 2010), pp. 3207–3220 (cit. on p. 20).

[23]Riccardo Coppola and Maurizio Morisio. „Connected Car: Technologies, Issues, Future Trends“. Version 469. In: *ACM Comput. Surv.* 49.3 (Oct. 2016), 46:1–46:36 (cit. on p. 1).

[24]Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. „The Cityscapes Dataset for Semantic Urban Scene Understanding“. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, June 2016, pp. 3213–3223 (cit. on p. 42).

[25]Joyce Dargay, Dermot Gately, and Martin Sommer. „Vehicle Ownership and Income Growth, Worldwide: 1960-2030“. Version 790. In: *The Energy Journal* 28.4 (2007), pp. 143–170 (cit. on p. 1).

[26]Jia Deng, Wei Dong, Richard Socher, et al. „Imagenet: A large-scale hierarchical image database“. In: 2009 (cit. on pp. 3, 10, 20, 24, 38, 59, 72, 81).

[28]Daniel J. Fagnant and Kara Kockelman. „Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations“. Version 55. In: *Transportation Research Part A: Policy and Practice* 77 (July 1, 2015), pp. 167–181 (cit. on p. 2).

[29]M. Fiore, J. Harri, F. Filali, and C. Bonnet. „Vehicular Mobility Simulation for VANETs“. In: 40th Annual Simulation Symposium (ANSS'07). Mar. 2007, pp. 301–309 (cit. on p. 1).

[30]Kay Fuerstenberg, Dirk T Fuerstenberg, Klaus C J Linzmeier, and Dietmayer J. „Pedestrian recognition and tracking of vehicles using a vehicle based multilayer laserscanner“. Version 824. In: (Feb. 2, 2018) (cit. on p. 2).

[31]K. Fukushima. „Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position“. Version 253. In: *Biological Cybernetics* 36.4 (1980), pp. 193–202 (cit. on p. 18).

[32]T. Gandhi and M. M. Trivedi. „Pedestrian Protection Systems: Issues, Survey, and Challenges“. Version 741. In: *IEEE Transactions on Intelligent Transportation Systems* 8.3 (Sept. 2007), pp. 413–430 (cit. on pp. 2, 3, 8).

[33]Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Google-Books-ID: hNwqBAAAQBAJ. Springer Science & Business Media, Dec. 6, 2012. 289 pp. (cit. on p. 32).

[38] Mandar Haldekar, Ashwinkumar Ganesan, and Tim Oates. „Identifying Spatial Relations in Images using Convolutional Neural Networks“. Version 911. In: (June 13, 2017) (cit. on p. 83).

[40] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd edition. New York, NY: Springer, 2016. 745 pp. (cit. on p. 12).

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. „Deep Residual Learning for Image Recognition“. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778 (cit. on pp. 19, 23, 44).

[42] Highways England. *Highways England Strategic Road Network*. Vision document PR129/17. Version 837. London, Dec. 2017 (cit. on pp. 1, 2).

[44] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. „Neural Networks for Machine Learning“. Lecture Presentation. Toronto, 2015 (cit. on p. 45).

[45] Sepp Hochreiter. „The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions“. Version 663. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6 (Apr. 1, 1998), pp. 107–116 (cit. on p. 21).

[46] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. „Densely Connected Convolutional Networks“. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE, July 2017, pp. 2261–2269 (cit. on p. 19).

[49] Sergey Ioffe and Christian Szegedy. „Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift“. In: ICML'15. Lille, France: JMLR.org, 2015, pp. 448–456 (cit. on p. 23).

[50] Aleksei Grigorevich Ivakhnenko and Valentin Grigorevich Lapa. *Cybernetic predicting devices,* OCLC: 23815433. New York: CCM Information Corp., 1965 (cit. on p. 15).

[51] M. Jeong, B. C. Ko, and J. Y. Nam. „Early Detection of Sudden Pedestrian Crossing for Safe Driving During Summer Nights“. Version 438. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.6 (June 2017), pp. 1368–1380 (cit. on p. 9).

[52] Maria Karatsoli, Martin Margreiter, and Matthias Spangler. „Bluetooth-based travel times for automatic incident detection A systematic description of the characteristics for traffic management purposes“. Version 697. In: *Transportation Research Procedia*. 3rd Conference on Sustainable Urban Mobility, 3rd CSUM 2016, 26 27 May 2016, Volos, Greece 24 (Jan. 1, 2017), pp. 204–211 (cit. on p. 7).

[53] Hirokatsu Kataoka, Teppei Suzuki, Shoko Oikawa, Yasuhiro Matsui, and Yutaka Satoh. „Drive Video Analysis for the Detection of Traffic Near-Miss Incidents“. Version 917. In: (Apr. 7, 2018) (cit. on pp. 9, 20).

[54] Kitae Kim, Slobodan Gutesa, Branislav Dimitrijevic, et al. „Performance Evaluation of Video Analytics for Traffic Incident Detection and Vehicle Counts Collection“. In: Intelligent Systems Reference Library. Springer, Cham, 2017, pp. 213–231 (cit. on p. 8).

[55] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. „ImageNet Classification with Deep Convolutional Neural Networks“. In: ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., 2012, pp. 1097–1105 (cit. on pp. 3, 18, 21).

[56] Anders Krogh and John A. Hertz. „A Simple Weight Decay Can Improve Generalization“. In: Morgan Kaufmann, 1992, pp. 950–957 (cit. on p. 25).

[57] Jens Krüger and Rüdiger Westermann. „Linear Algebra Operators for GPU Implementation of Numerical Algorithms“. In: SIGGRAPH 2003. Vol. 22. ACM, 2003, pp. 908–916 (cit. on p. 21).

[58] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. „Gradient-based learning applied to document recognition“. Version 646. In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324 (cit. on pp. 18–21).

[59] Dan Levi, Noa Garnett, and Ethan Fetaya. „StixelNet: A Deep Convolutional Network for Obstacle Detection and Road Segmentation“. In: Jan. 1, 2015, pp. 109.1–109.12 (cit. on pp. 9, 20).

[62] Laurens van der Maaten and Geoffrey E. Hinton. „Visualizing Data using t-SNE“. Version 608. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605 (cit. on p. 28).

[63] Peter Martin, Joseph Perrin, Blake Hansen, Ryan Krump, and Dan Moore. *Incident Detection Algorithm Evaluation*. MPC-01-122. Version 693. Utah, Mar. 2001 (cit. on pp. 7, 8).

[64] Damien Matti, Hazim Kemal Ekenel, and Jean-Philippe Thiran. „Combining LiDAR space clustering and convolutional neural networks for pedestrian detection“. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). Lecce, Italy: IEEE, Aug. 2017, pp. 1–6 (cit. on p. 9).

[68] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. „WordNet: An on-line lexical database“. Version 869. In: *International Journal of Lexicography* 3 (1990), pp. 235–244 (cit. on pp. 10, 72).

[69] J. Miller. „Vehicle-to-vehicle-to-infrastructure (V2V2I) intelligent transportation system architecture“. In: 2008 IEEE Intelligent Vehicles Symposium. June 2008, pp. 715–720 (cit. on p. 1).

[70] Tom Mitchell. *Rosenblatt's Perceptron*. Version 533. Oct. 24, 2012 (cit. on p. 14).

[71] V. Monga and B. L. Evans. „Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs“. Version 856. In: *IEEE Transactions on Image Processing* 15.11 (Nov. 2006), pp. 3452–3465 (cit. on p. 73).

[73] Keiller Nogueira, Otvio A.B. Penatti, and Jefersson A. dos Santos. „Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification“. Version 924. In: *Pattern Recogn.* 61 (C Jan. 2017), pp. 539–556 (cit. on p. 26).

[74] S. Park and C. C. Zou. „Reliable Traffic Information Propagation in Vehicular Ad-Hoc Networks“. In: 2008 IEEE Sarnoff Symposium. Apr. 2008, pp. 1–6 (cit. on p. 6).

[75] P.B. Farradyne. *Traffic Incident Management Handbook*. Google-Books-ID: j3MsAQAAMAAJ. U.S. Department of Transportation, ITS Joint Program Office, 2000. 182 pp. (cit. on p. 5).

[76]Nick Peluffo. *Strategic Road Network Statistics*. Governmental Statistics. Version 836. Jan. 2015 (cit. on p. 1).

[77]Luis Perez and Jason Wang. „The Effectiveness of Data Augmentation in Image Classification using Deep Learning“. Version 926. In: (Dec. 13, 2017) (cit. on p. 20).

[80]Raúl Rojas. *Neural Networks: A Systematic Introduction*. Berlin, Heidelberg: Springer-Verlag, 1996 (cit. on pp. 16, 17).

[81]F. Rosenblatt. *The Perceptron: A Probabilistic Model for Information Storage and Organization*. 1958 (cit. on p. 14).

[82]David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. „Learning representations by back-propagating errors“. Version 543. In: *Nature* 323.6088 (Oct. 1986), pp. 533–536 (cit. on p. 15).

[83]T S Huang. „Computer Vision: Evolution and Promise“. Version 522. In: (July 27, 2018) (cit. on p. 12).

[84]Mark S. Nixon and A.S. Aguado. „Feature Extraction & Image Processing for Computer Vision“. Version 521. In: *Feature Extraction & Image Processing for Computer Vision* (Dec. 1, 2012) (cit. on p. 13).

[85]A. Salas, P. Georgakis, and Y. Petalas. „Incident detection using data from social media“. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). Oct. 2017, pp. 751–755 (cit. on p. 7).

[86]Khaled Saleh, Mohammed Hossny, and Saeid Nahavandi. „Kangaroo vehicle collision detection using deep semantic segmentation convolutional neural network“. Version 419. In: *DICTA 2016 : Proceedings of the IEEE International Conference on Digital Image Computing: Techniques and Applications* (Jan. 1, 2016) (cit. on pp. 9, 20).

[88]Brandon Schoettle and Michael Sivak. „A survey of public opinion about autonomous and self-driving vehicles in the US, the UK, and Australia“. Version 51. In: (2014) (cit. on p. 3).

[91]H. Shao, Z. Zhang, and K. Li. „Research on water hazard detection based on line structured light sensor for long-distance all day“. In: 2015 IEEE International Conference on Mechatronics and Automation (ICMA). Aug. 2015, pp. 1785–1789 (cit. on p. 9).

[92]Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tang. „On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima“. Version 918. In: (Sept. 15, 2016) (cit. on p. 25).

[93]Gary Silberg and Richard Wallace. *Self-driving cars: The next revolution*. Version 228. 2012 (cit. on p. 2).

[94]K. Simonyan. „Deep inside convolutional networks: visualising image classification models and saliency maps“. Version 927. In: (Jan. 1, 2014) (cit. on pp. 26, 27).

[98]C. Szegedy, Wei Liu, Yangqing Jia, et al. „Going deeper with convolutions“. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA: IEEE, June 2015, pp. 1–9 (cit. on pp. 19, 21).

[99]Yichuan Tang. „Deep Learning using Linear Support Vector Machines“. Version 939. In: *arXiv:1306.0239 [cs, stat]* (June 2, 2013). arXiv: 1306.0239 (cit. on p. 19).

[103]C.j. Van Rijsbergen. „Foundation of evaluation“. Version 952. In: *Journal of Documentation* 30.4 (Apr. 1, 1974), pp. 365–373 (cit. on p. 44).

[106]Martin Wattenberg, Fernanda Viégas, and Ian Johnson. „How to Use t-SNE Effectively“. Version 854. In: *Distill* 1.10 (Oct. 13, 2016), e2 (cit. on pp. 29, 47).

[107]Thomas Weise. *Global Optimization Algorithms - Theory and Applications*. 2nd ed. self-published, 2009 (cit. on p. 16).

[108]Paul John Werbos. *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*. New York, NY: Adaptive and Learning Systems, Feb. 11, 1994. 319 pp. (cit. on p. 15).

[110]Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. „LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop“. Version 941. In: (June 10, 2015) (cit. on pp. 11, 38).

[112]Dehuai Zeng and Jianmin Xu. „Data Fusion for Traffic Incident Detection Using D-S Evidence Theory with Probabilistic SVMs“. In: World Congress on Engineering and Computer Science. 2010 (cit. on p. 7).

[113]Kun Zhang and Michael A. P. Taylor. „Towards Transferable Incident Detection Algorithms“. Version 690. In: *Journal of the Eastern Asia Society for Transportation Studies* 6 (2005), pp. 2263–2274 (cit. on p. 7).

[114]S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. „Towards Reaching Human Performance in Pedestrian Detection“. Version 742. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (Apr. 2018), pp. 973–986 (cit. on p. 9).

[115]Bolei Zhou. *CAM: Class Activation Mapping*. Version 853. original-date: 2016-04-11T21:59:28Z. Aug. 21, 2018 (cit. on p. 47).

[116]Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. „Learning Deep Features for Discriminative Localization“. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, June 2016, pp. 2921–2929 (cit. on pp. 27, 47).

[117]Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. „Object detectors emerge in Deep Scene CNNs“. Version 946. In: (Dec. 21, 2014) (cit. on p. 27).

[118]Depu Zhou. „Real-time Animal Detection System for Intelligent Vehicles“. Thesis. 2014 (cit. on p. 9).

[119]Zhi-Hua Zhou. „A brief introduction to weakly supervised learning“. Version 957. In: *National Science Review* 5.1 (Jan. 1, 2018), pp. 44–53 (cit. on p. 63).

Web Sources

[1]12019. *Free Image on Pixabay - Landscape, Forest, Trees, Woods*. Pixabay.com. Version 949. July 29, 2017. URL: `www.pixabay.com/en/landscape-forest-trees-woods-road-2550393/` (visited on Aug. 5, 2018) (cit. on pp. 36, 64).

[2] Sam Abuelsamid. *BMW, HERE And Mobileye Team Up To Crowd-Source HD Maps For Self-Driving*. Forbes. Version 229. 2018. URL: https://www.forbes.com/sites/samabuelsamid/2017/02/21/bmw-here-and-mobileye-team-up-to-crowd-source-hd-maps-for-self-driving/ (visited on Feb. 2, 2018) (cit. on p. 2).

[6] Amazon. *Amazon Mechanical Turk*. Amazon Mechanical Turk. Version 871. 2018. URL: https://www.mturk.com/ (visited on Aug. 23, 2018) (cit. on p. 11).

[8] Arrive Alive. *KZN: R34 / John Ross road collapses outside Richards Bay during peak traffic time. https://shar.es/1ShH4că @Netcare911_sa #ArriveAlivepic.twitter.com/h2yJXv4qLP*. @_ArriveAlive. Version 582. Aug. 25, 2017. URL: https://twitter.com/_ArriveAlive/status/901132427522965508/photo/1?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E901132427522965508&ref_url=https%3A%2F%2Fwww.timeslive.co.za%2Fnews%2Fsouth-africa%2F2017-08-26-woman-has-narrow-escape-as-road-collapses%2F (visited on Aug. 5, 2018) (cit. on pp. 36, 64).

[9] Terryl Asla. *Brownsville Highway reopened after being closed due to fallen tree*. Kitsap Daily News. Version 575. Jan. 20, 2017. URL: https://www.kitsapdailynews.com/news/brownsville-highway-closed-due-to-fallen-tree/ (visited on Aug. 5, 2018) (cit. on pp. 36, 64).

[10] Rodrigo Benenson. *Classification datasets results*. What is the class of this image ? Version 474. 2016. URL: http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#43494641522d3130 (visited on Feb. 17, 2018) (cit. on p. 20).

[27] David Dixon. *Roadworks and Diversion at Magheralahan*. Geograph - Photograph Every Grid Square. Version 719. Sept. 26, 2017. URL: https://www.geograph.ie/photo/5682971 (visited on Aug. 16, 2018) (cit. on p. 6).

[34] *Geograph*. Geograph - Photograph Every Grid Square. Version 453. 2018. URL: http://www.geograph.org.uk/ (visited on Feb. 17, 2018) (cit. on p. 42).

[35] Geograph Project. *Geograph*. Version 448. URL: https://www.geograph.org.uk/ (visited on May 27, 2018) (cit. on p. 38).

[36] Given Up. *Geograph:: Bentley Road (C) Given Up*. Geograph - Photograph Every Grid Square. Version 850. Mar. 23, 2008. URL: http://www.geograph.org.uk/photo/740204 (visited on Aug. 20, 2018) (cit. on p. 43).

[37] Google. *Google Colaboratory*. Google Colaboratory. Version 487. 2018. URL: https://colab.research.google.com (visited on July 13, 2018) (cit. on pp. 38, 46).

[39] Greta Harrison. *Can hackers turn off the lights?* Pursuit. Version 876. Jan. 18, 2016. URL: https://pursuit.unimelb.edu.au/articles/can-hackers-turn-off-the-lights (visited on Aug. 27, 2018) (cit. on p. 6).

[43] Highways England. *Smart motorways programme*. Version 788. July 14, 2016. URL: http://www.highways.gov.uk/smart-motorways-programme/ (visited on Aug. 17, 2018) (cit. on p. 1).

[47] Iceland24. *Winter driving tips in Iceland! - Icelandic roads*. Iceland24 - Iceland Travel and Info Guide. Version 863. Nov. 2, 2017. URL: http://www.iceland24blog.com/2017/11/winter-driving-tips-in-iceland.html (visited on Aug. 23, 2018) (cit. on p. 75).

[48]IHS Markit. *Autonomous Vehicle Sales Forecast 2018 - SupplierInsight*. SupplierInsight. Version 795. 2018. URL: https://supplierinsight.ihsmarkit.com/shop/product/5001816/autonomous-vehicle-sales-forecast-and-report (visited on Aug. 17, 2018) (cit. on pp. 1, 2).

[60]Fei-Fei Li, Justin Johnson, and Serena Yeung. *CS231n Convolutional Neural Networks for Visual Recognition*. CS231n: Convolutional Neural Networks for Visual Recognition. Version 507. 2018. URL: http://cs231n.github.io/neural-networks-1/#actfun (visited on July 20, 2018) (cit. on p. 19).

[61]Fei-Fei Li, Justin Johnson, and Serena Yeung. *Stanford University CS231n: Convolutional Neural Networks for Visual Recognition*. Version 679. 2018. URL: http://cs231n.github.io/neural-networks-3/#hyper (visited on Jan. 30, 2018) (cit. on p. 24).

[65]MaviccPRP@web.studio. *A Neural Network from scratch in just a few Lines of Python Code*. MaviccPRP@web.studio. Version 542. Apr. 13, 2017. URL: https://maviccprp.github.io/a-neural-network-from-scratch-in-just-a-few-lines-of-python-code/ (visited on July 27, 2018) (cit. on p. 17).

[66]Matt McFarland. *Uber shuts down self-driving operations in Arizona*. CNNMoney. Version 809. May 23, 2018. URL: https://money.cnn.com/2018/05/23/technology/uber-arizona-self-driving/index.html (visited on Aug. 17, 2018) (cit. on p. 3).

[67]Microsoft. *Bing Images Search API v7 Reference | Microsoft Docs*. Image Search API v7 Reference. Version 451. URL: https://docs.microsoft.com/en-us/rest/api/cognitiveservices/bing-images-api-v7-reference (visited on May 27, 2018) (cit. on p. 38).

[72]Navigant Research. *Navigant Research Leaderboard: Automated Driving Vehicles*. Navigant. Version 813. 2018. URL: http://62.nl.dealer-preview.co/SiteAssets/LB-AV-17-Navigant-Research_FINAL.pdf (visited on Aug. 17, 2018) (cit. on p. 1).

[78]Prabhu. *CNN Architectures LeNet, AlexNet, VGG, GoogLeNet and ResNet*. Medium. Version 661. Mar. 15, 2018. URL: https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848 (visited on Aug. 14, 2018) (cit. on p. 22).

[79]Prawny. *Free Image on Pixabay - Fowey, Village, Cornwall, Street*. Pixabay.com. Version 949. Feb. 14, 2014. URL: www.pixabay.com/en/fowey-village-cornwall-street-road-268652/ (visited on Aug. 5, 2018) (cit. on pp. 36, 64).

[87]Sandy Spring Volunteer Fire Department. *SSVFD Special Operations Put to Work During Recent Flash Flood*. Sandy Spring Volunteer Fire Department. Version 719. May 25, 2017. URL: https://www.ssvfd.org/ssvfd-special-operations-put-work-recent-flash-flood/ (visited on Aug. 16, 2018) (cit. on p. 6).

[89]Scikit. *sklearn.manifold.TSNE scikit-learn 0.19.2 documentation*. scikit-learn.org. Version 705. 2018. URL: http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html (visited on Aug. 15, 2018) (cit. on pp. 29, 47).

[90]Daniel Seita. *BDD100K: A Large-scale Diverse Driving Video Database*. The Berkeley Artificial Intelligence Research Blog. Version 511. May 30, 2018. URL: http://bair.berkeley.edu/blog/2018/05/30/bdd/ (visited on June 4, 2018) (cit. on p. 41).

[95] Skeeze. *Free Image on Pixabay - Winter, Wonderland, Landscape*. Pixabay.com. Version 949. Feb. 12, 2006. URL: www.pixabay.com/en/winter-wonderland-landscape-scenic-581101/ (visited on Aug. 5, 2018) (cit. on pp. 36, 64).

[96] Society of Automotive Engineers. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles (Standard)*. Version 375. Aug. 21, 2017. URL: http://www.roadsafetyobservatory.com/Evidence/Details/11728 (visited on Feb. 19, 2018) (cit. on p. 1).

[97] Stanford Vision Lab and Princeton University. *ImageNet*. ImageNet. Version 587. 2016. URL: http://www.image-net.org/search?q=animal (visited on Aug. 7, 2018) (cit. on p. 61).

[100] Telegra. *Traffic Video Analysis - Automatic Video Incident Detection| Telegra | Telegra | Telegra*. Telegra Europe. Version 728. URL: https://www.telegra-europe.com/solutions/solution-54 (visited on Aug. 16, 2018) (cit. on p. 8).

[101] Ian Thompson. *Geograph:: Minor Road Near Skeoch (C) Iain Thompson*. Geograph - Photograph Every Grid Square. Version 845. Dec. 7, 2010. URL: http://www.geograph.org.uk/photo/2193653 (visited on Aug. 20, 2018) (cit. on p. 43).

[102] vainodesositis. *Free Image on Pixabay - Car Accident, Fire, Street*. Pixabay.com. Version 950. Sept. 14, 2017. URL: www.pixabay.com/en/car-accident-fire-street-accident-2789841/ (visited on Aug. 5, 2018) (cit. on pp. 36, 64).

[104] Petar Velikovi. *Deep learning for complete beginners: convolutional neural networks with keras*. Deep learning for complete beginners: convolutional neural networks with keras. Version 554. Mar. 20, 2017. URL: http://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html (visited on July 30, 2018) (cit. on p. 19).

[105] Virginia Department of Traffic Salem. *Rock slide in Botetourt County causes road closure*. WSET. Version 951. Apr. 6, 2017. URL: https://twitter.com/VaDOTSalem/status/850051876997484544?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed&ref_url=http%3A%2F%2Fwset.com%2Fnews%2Flocal%2Frock-slide-in-botetourt-county-causes-road-closure (visited on Aug. 5, 2018) (cit. on pp. 36, 64).

[109] Yahoo. *Flickr Services API documentation*. The App Garden. Version 451. URL: https://www.flickr.com/services/api/ (visited on May 27, 2018) (cit. on p. 38).

[111] Nanda Yugandar. *What is the VGG neural network? - Quora*. What is the VGG neural network? Version 665. Aug. 5, 2018. URL: https://www.quora.com/What-is-the-VGG-neural-network (visited on Aug. 14, 2018) (cit. on p. 22).