Journal of Intelligent Learning Systems and Applications, 2024, 16, 107-142
https://www.scirp.org/journal/jilsa
ISSN Online: 2150-8410
ISSN Print: 2150-8402

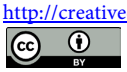
Real Time Thermal Image Based Machine Learning Approach for Early Collision Avoidance System of Snowplows

Fletcher Wadsworth[1], Suresh S. Muknahallipatna[1*], Khaled Ksaibati[2]

[1]Department of Electrical Engineering and Computer Science, University of Wyoming, Laramie, WY, USA
[2]Department of Civil and Architectural Engineering, University of Wyoming, Laramie, WY, USA
Email: fwadswor@uwyo.edu, *sureshm@uwyo.edu, khaled@uwyo.edu

How to cite this paper: Wadsworth, F., Muknahallipatna, S.S. and Ksaibati, K. (2024) Real Time Thermal Image Based Machine Learning Approach for Early Collision Avoidance System of Snowplows. *Journal of Intelligent Learning Systems and Applications*, **16**, 107-142.
https://doi.org/10.4236/jilsa.2024.162008

Received: April 9, 2024
Accepted: May 26, 2024
Published: May 29, 2024



Abstract

In an effort to reduce vehicle collisions with snowplows in poor weather conditions, this paper details the development of a real time thermal image based machine learning approach to an early collision avoidance system for snowplows, which intends to detect and estimate the distance of trailing vehicles. Due to the operational conditions of snowplows, which include heavy-blowing snow, traditional optical sensors like LiDAR and visible spectrum cameras have reduced effectiveness in detecting objects in such environments. Thus, we propose using a thermal infrared camera as the primary sensor along with machine learning algorithms. First, we curate a large dataset of thermal images of vehicles in heavy snow conditions. Using the curated dataset, two machine-learning models based on the modified ResNet architectures were trained to detect and estimate the trailing vehicle distance using real-time thermal images. The trained detection network was capable of detecting trailing vehicles 99.0% of the time at 1500.0 ft distance from the snowplow. The trained trailing distance network was capable of estimating distance with an average estimation error of 10.70 ft. The inference performance of the trained models is discussed, along with the interpretation of the performance.

Keywords

Convolutional Neural Networks, Residual Networks, Object Detection, Image Processing, Thermal Imaging

1. Introduction

Snowplows operate in hazardous road conditions to remove or reduce snow accumulation on roads. Snow removal is typically managed by a state's Depart-

ment of Transportation (DOT) for heavily trafficked roads like interstates and highways in the US. Snow often falls while plows remove snow, reducing the visibility of the snowplow operators and consumer vehicles. Moreover, when the snowplow moves with its blade depressed into a volume of snow, it causes a flurry of displaced snow surrounding the plow, known as the snow plume, further reducing the snowplow operator's visibility and the visibility of both approaching and following vehicles. These conditions result in collisions with snowplows, and recently, in a single day, multiple collisions [1] experienced by snowplows were reported. Zockie *et al.* [2] analyzed crash reports from 2012 to 2017 in Michigan UD-10 and determined that 1354 crashes involved snowplows. Research conducted at Virginia Tech [3] showed that 22.8% of crashes involving snowplows were due to inattention/misjudgment by other drivers. Haq *et al.* [4] analyzed the snowplow-related injury severity along mountainous roadways in Wyoming and determined a significant number of crashes occur due to the slow speeds of snowplows. These accidents are often caused by vehicles approaching the snowplow from the rear due to poor visibility of the snowplow and the inability to stop in time. Furthermore, vehicles attempting to pass a snowplow cause head-on collisions with vehicles traveling in the opposite direction, leading to a cascading accident involving the snowplow. These accidents involving snowplows have multiple consequences, such as human physical impairments, death, and economic costs. Typically, the time and cost to repair a snowplow involved in a rear collision is two to three months and $100,000. Furthermore, DOTs have limited snowplows, so losing a snowplow due to a collision impedes snow removal on interstates and highways, resulting in perpetuating poor road conditions and, subsequently, more accidents. Thus, a collision avoidance system capable of alerting snowplow operators to approaching vehicles from the rear will allow the snowplow operators to alert (using warning lights and horns) the trailing vehicles about the presence of a snowplow ahead, raise the blade to reduce the intensity of the snow plume, and/or move on to the shoulder of the road to avoid possible collisions.

However, heavy snowfall and blowing snow impede detecting objects with visual sensors like visible spectrum (RGB) cameras and LiDAR. An image taken from an RGB camera in heavy snowfall and blowing snow conditions will have a reduced perception of the details of the image. Falling and blowing snow first occludes the view of an image sensor due to the reflectivity of snow in the visible spectrum; subsequently, the increasing density of falling snow creates opaqueness in the visible wavelengths [5] for the image sensor. Similarly, LiDAR has reduced range and efficacy in snow conditions due to light reflection at the wavelengths typical for LiDAR sensors (~905 nm) [6]. While the reflection of light due to snow particles will result in false vehicle detection, the opaqueness will prevent vehicle detection. These issues with visible cameras and LiDAR in snowy conditions make them unsuitable for detecting vehicles trailing a snowplow. In this paper, we implement a real-time collision avoidance system for a snowplow consisting of two machine learning models trained to detect trailing vehicles and

their distances from the snowplow using thermal images obtained from the 8 - 14 μm spectral band thermal infrared camera. We use a thermal camera as the sensor instead of a visible spectrum camera and LiDAR, since a thermal camera capturing heat differentials in its pixel intensities should be able to capture the heat differential between a vehicle and the environment.

This paper is organized as follows: Section 2 provides a brief summary of the related work. In Section 3, thermal image data collection and preprocessing is presented. Section 4 discusses architecture choices for the ML models. In Section 5, the results of the models' training are presented. Section 6 presents the performance of the trained models, particularly on samples which were predicted poorly. Finally, section 7 discusses the conclusion of this research, as well as future research directions.

2. Related Work

Continuous improvements in autonomous vehicle research have resulted in several obstacle detection and avoidance approaches and have demonstrated remarkable success in practical implementation, as reviewed by Siddiqui *et al.* [7]. These approaches typically utilize sensors such as visible spectrum cameras, LiDAR, and radar sensors to collect variations of visual data of the vehicle's surroundings and subject the data to algorithms that parse the data into information of the relative location of detected obstacles. As the capability of edge computing devices improves and available training data continuously increases, these algorithms are trending toward deep learning models.

Much research has been done on vehicle detection in the context of autonomous driving. However, relevant research using thermal cameras for obstacle detection is more sparse. Bhadoriya *et al.* [8], in their research, have collected thermal images manually in adverse weather conditions to train a YOLO-based vehicle detection model. They have demonstrated the efficiency of using thermal images with the YOLO model in a simulator. They have used radar to measure the distance of other vehicles from the autonomous vehicle. Measuring the vehicle distances using the radar is appropriate since they have not considered snow as one of the adverse weather condition scenarios. Furthermore, considering only objects in the direction along a fixed line of the autonomous vehicle makes distance measurements using the radar accurate.

Alhamaddi *et al.* [9] have trained a transfer learning-based vehicle detection model using thermal image data in adverse conditions. The thermal images of vehicles were collected only in heavy fog formation and the dataset consisted of only 70 gray scale images. This research demonstrated that transfer learning with a large pre-trained architecture could be used to reduce computational costs and the burden of architectural searches.

Lu *et al.* [10] have improved the yolov3-tiny model and applied it to thermal vehicle image data for object detection. They have used the FLIR ADAS Dataset which contains thermal images of fifteen different vehicle types in adverse

weather conditions such as total darkness, fog, smoke, rain, and glare. This research focuses on modifying the yolov3-tiny model by adding a detection layer to detect multiple small objects closer to each other. However, this work did not address detecting objects in snowy conditions and detecting the distance.

Kang *et al.* [11] collected nighttime thermal data in urban and suburban traffic scenarios and compared the efficacy of several lightweight CNN models for classifying four different vehicle types: cars, buses, trucks, and vans. This work demonstrates the efficiency of small CNNs using thermal image data for a simple classification task. In this work, the authors have demonstrated that network efficiency is a critical component of real-time vehicle detection; models must have sufficiently low inference time and memory usage on resource-constrained edge devices to maintain high inference throughput and ensure that the most recent predictions are always available. Despite the important insights in using thermal data for vehicle classifications, thermal data was not collected in snow conditions and was limited to a distance of 50 m. Additionally, no distance estimation techniques are addressed.

Research specifically concerning thermal image vehicle detection in heavy snow conditions is scarce. Han and Hu [12] explored vehicle detection with thermal and visible spectrum imaging in the context of traffic surveillance cameras. They used RGB images augmented with thermal grayscale images in a dual input faster RCNN network and trained with rain and snow condition data. However, the fixed nature of traffic monitoring cameras makes vehicles always appear against the same background at a small range of distances and do not reflect the high variance environments seen by cameras attached to moving vehicles. As with most other research in this area, the estimation of vehicle distance was not examined.

In all of the above research, none of the thermal data used are in snow conditions. In all this research, object detection concerns an autonomous vehicle in a fixed line of sight and small distances. To measure the distance of the detected object from an autonomous vehicle, either a radar or LiDAR is used. Our research addresses object detection at large distances over 1000 ft, varying lines of sight, and in heavy snowfall conditions. Since radars and LiDAR do not function accurately in snowy conditions to detect distances, we propose creating a custom dataset of thermal images labeled with the distance and training two deep learning models; one for detecting the objects and the other for estimating the distance.

3. Thermal Data

Curation As with any machine learning application, a sufficiently large, varied, and high-quality dataset is key to ensuring the deployment performance of a trained model.

3.1. Data Collection

We have used the FLIR ADK thermal infrared camera to collect thermal images

of vehicles during snow events, such as in Figure 1.

Thermal images and corresponding distance labels are collected ad hoc during heavy snow events with different vehicle makes and models. We collect the thermal image data and distance using two vehicles, identified as the *leader* and *follower*, traveling on interstates, highways, and other roads maintained by the Wyoming Department of Transportation. In addition to the leader vehicle equipped with the thermal camera, both have GPS and LoRa transmitters and receivers. Constant communication of the GPS coordinates of the leader and follower vehicles between them is maintained using the LoRa transmitter/receivers to ensure a desired separation distance. The leader vehicle simulates a snowplow, while the follower vehicle simulates trailing consumer vehicles. During the data collection process, we collect thermal images using different makes and models of the follower vehicles to capture the variety of real traffic driving characteristics accurately.

During training data collection, the follower vehicle drives behind the leader and varies its distance from the leader vehicle by 100 ft to over 1000 ft. Thermal images of the follower vehicle are captured continuously from the rear of the leader vehicle, and each collected image is annotated with the distance separating them determined using the GPS information of both vehicles. The thermal images with other vehicles between the leader and follower vehicles are discarded from the distance estimation data set. Figure 2 illustrates this process.

A thermal image contains pixel intensities that correlate to the heat emitted from areas in the image scene. In snowy conditions, due to occlusions from snowflakes, a vehicle will not be discernible in the image from an RGB camera,

Figure 1. Vehicle thermal image at 100 ft distance from thermal camera.

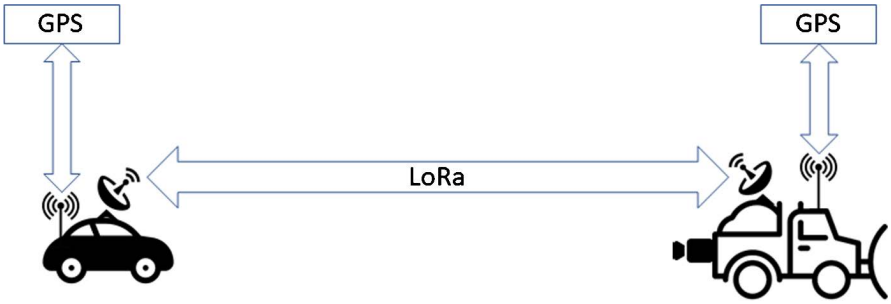


Figure 2. Data collection procedure diagram.

particularly a vehicle at a significant distance from the RGB camera. However, in snowy conditions, the heat signature of a vehicle can be detected as it is pronounced significantly due to the ambient temperature typically being below freezing (32 F). We propose to collect the heat signature data using a thermal image, where the pixel intensity in the thermal image correlates to the heat emitted in the image scene. Due to the inherent issues with infrared thermography listed below, thermal images are generally noisier and of worse quality than RGB images.

- The intensity of infrared radiation emitted by objects is weaker than visible light due to the lower frequency of the infrared band, resulting in a lower intensity differential between pixel regions of objects of different temperatures. Lower cost commercial thermal cameras lack the costly high-sensitivity sensors to capture the weak signal differential, reducing the capacity of these images to express temperature gradients at a low resolution.
- The diffraction limited resolution $\theta \propto f^{-1}$ is proportional to the wavelength of detected light [13]. Since infrared light has a longer wavelength than visible spectrum light, larger pixel sites are required on the sensor array, decreasing image resolution.
- To reduce the accumulation of snow on the lens of a thermal camera, the thermal camera is equipped with a heating element. The resulting Johnson (temperature) noise, and to a lesser extent the noise caused by other camera electronics ($1/f$ noise, fixed pattern noise, etc.) cause images obtained from thermal cameras to exhibit characteristic noise patterns [14].

To ameliorate this noise and increase the visibility of areas of interest in thermal images, we preprocess the thermal images before they are used as input to the perception models.

3.2. Thermal Image Preprocessing

3.2.1. Normalization

First, we quantize each pixel in the thermal image to integers in the range [0, 255] through the normalization process to maximize the contrast between features in the image. The normalization process involves re-scaling the raw data from the thermal image to the maximum dynamic range [0, 255] as given by the Equation (1).

$$\hat{I} = \left(I - I_{\min}\right)\frac{\beta - \alpha}{I_{\max} - I_{\min}} + \beta \tag{1}$$

where $I_{\min}$ is the smallest pixel intensity value in the image, $I_{\max}$ is the largest, $\alpha = 0$, and $\beta = 255$.

3.2.2. Denoising

As mentioned above, thermal cameras are inherently noisy. A denoising algorithm is deployed on the normalized image to reduce this effect. Denoising typically involves the application of a Gaussian or median blur filter to replace a pixel value with an average of the pixel values around it. However, this technique will cause the unwanted smoothing of any image area with a periodic or repeating structure. The spatial coherency issue degrades the visual quality of any repeated patterns in an image denoised by blurring methods and can be remedied using the *Non-Local Means (NLM) Denoising* [15] algorithm. The NLM denoising with a patch-wise implementation uses a research window of a large neighborhood of pixels or patches in the vicinity of the patch under investigation and, within this research window, searches for patches that resemble the target patch. The patches in the research window are weighted to reflect their resemblance to the target patch, and a weighted sum of the patches is computed to replace the target patch.

Consider a normalized thermal image *u*. The denoising of this image is given by the Equation (2),

$$\hat{B} = \frac{1}{C(p)}\sum_{Q\in R} u(q)w(B,Q),\ C(p) = \sum_{Q\in R} w(B,Q), \tag{2}$$

where $B = B(p,f)$ indicates the target patch under consideration, centered at p with side length $2f+1$, $R = R(p,r)$ indicates the rectangular research window, centered at p with a side length of $2r+1$, and $Q = Q(q,f) \in R(p,r)$ indicates each patch within the research window of the same size as the target patch, each patch centered at q. The weights $w(B,Q)$ are calculated using an exponential kernel given by the Equation (3),

$$w(B,Q) = \exp\left[-\frac{d^2}{h^2}\right], \tag{3}$$

where d is the Euclidian distance between the target patch B and test patch Q, and h is a tunable filter strength parameter. This weight value is set to 1 (*i.e.*, d^2 is set to 0) if the patches are sufficiently close, a threshold set per implementation based on the variance of the image noise.

The research window size r, patch size f, and filter strength h must be chosen at design time. Increasing the research window size allows access to a larger neighborhood of potential patches for more global denoising but increases computation time. Increasing the patch size is desirable when the image noise has a large variance; however, this requires a larger research window to find more similar patches successfully. High filter strength h results in the weights

concentrating on very similar patches, which removes noise better but is more likely to degrade image details. Without a noise model, optimal parameter values cannot be chosen mathematically and thus were tuned by observing the denoising performance on the thermal image dataset. The denoising tuned parameters for the normalized input thermal images were determined as $r = 10$, $f = 3$, and $h = 10$.

3.2.3. Dilation

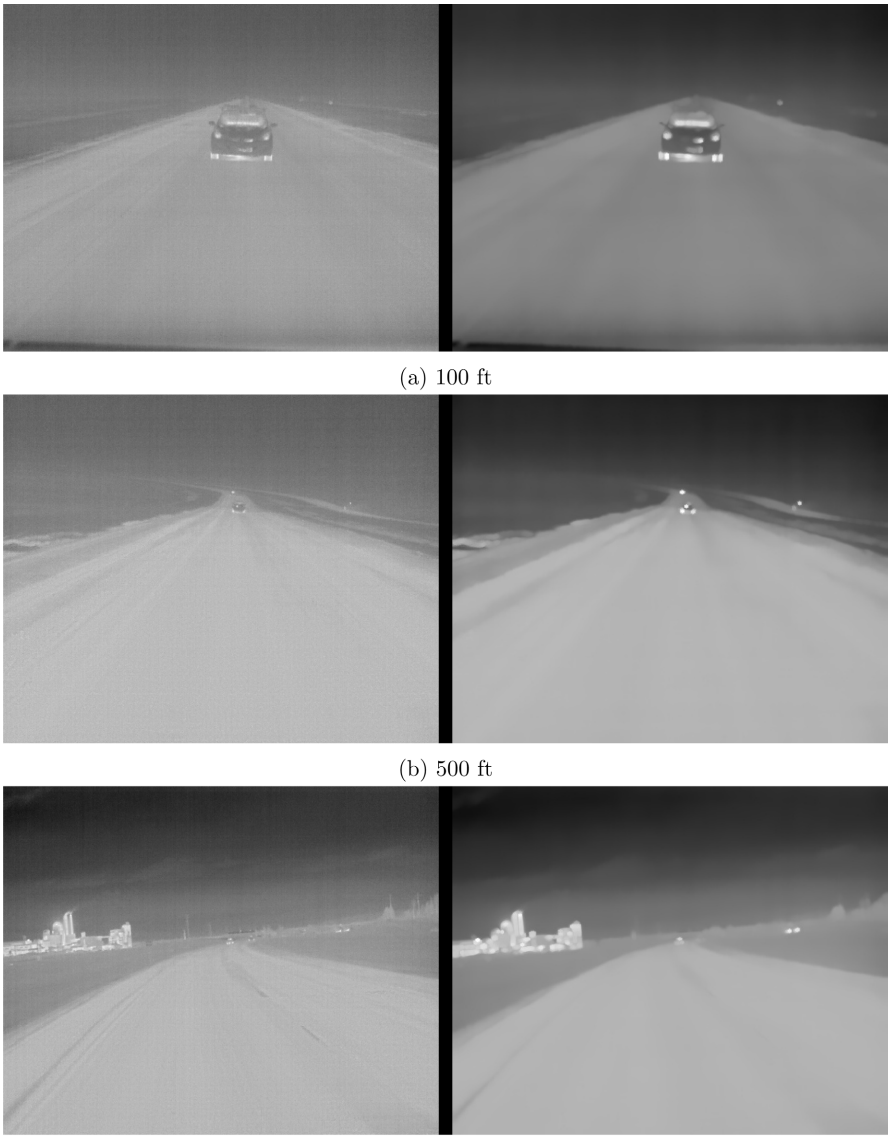
Since trailing vehicles can be at a long distance from the thermal camera, objects of interest may be represented only by a few pixels, resulting in the desired image features being low resolution and difficult to detect. Image dilation is a morphological operation that changes shapes' boundaries by applying a structuring element to an input image. The image dilation adds higher intensities to boundary pixels of shapes in the input image. In essence, this operation causes bright regions in the image to grow, accentuating the low-resolution features represented by high pixel intensities. We propose dilation as a technique to emphasize vehicles in thermal images.

Dilation involves convolving the input image I with a structuring element or kernel C, which was chosen as a 5 × 5 circular kernel. A circular kernel was chosen to avoid introducing sharp corner artifacts in the output image that can happen with a rectangular kernel. The kernel size was selected as a middle ground between insufficient feature accentuation (as with a 3 × 3 kernel) and too much degradation of the vehicle structure at closer distances (with 7 × 7 and larger kernels), chosen by experimentation and visual analysis. As C is convolved over the image I, the maximum pixel value that overlaps with the kernel is found, and the value of the pixel that anchors the kernel is replaced with this maximum value. This has the effect of making regions of high pixel intensity appear larger in the output image. Sample thermal images and their corresponding preprocessed images are shown in **Figure 3**.

In **Figure 3**, the increase in contrast between dark regions like the sky and lighter regions like the road and vehicle features due to normalizing can be observed. The effects of denoising can also be seen where regions of similar intensity are more uniform in the output image. Finally, key features of the trailing vehicle, indicated by the brightest pixel intensities in the image corresponding to the windshield, headlights, and tires, appear larger and more distinct after dilation. Furthermore, at distances of 500 ft and 1000 ft, it can be observed that preprocessing significantly increases the contrast between the vehicle and the background.

4. Deep Learning Detection and Trailing Distance Models

The snowplow operator requires two vital pieces of information, the awareness of a vehicle trailing the snowplow and the distance between the snowplow and the trailing vehicle, to operate the snowplow safely and avoid rear-end collisions. Using these two vital pieces of information in real-time, a snowplow operator



(a) 100 ft

(b) 500 ft

(c) 1000 ft

Figure 3. Comparison of thermal images at varying distances before and after preprocessing.

can turn on warning lights/horns to warn the trailing vehicle driver of the presence of a snowplow. We propose to detect the presence of a trailing vehicle and the trailing distance using two deep-learning models in cascade, as shown in Figure 4. Since an energy-constrained edge device such as NVIDIA Jetson is used to perform real-time inference to predict the distance between the trailing vehicle and the snowplow, it is necessary to minimize the computations for inference. Every image from the thermal camera at the frame rate of 30 samples/second may not have a vehicle, and therefore, predicting the trailing distance using each thermal image frame would result in unnecessary computations, increasing energy consumption and potentially reducing throughput. In Figure 4, it can be observed that the Detection Model (DM) is first used to

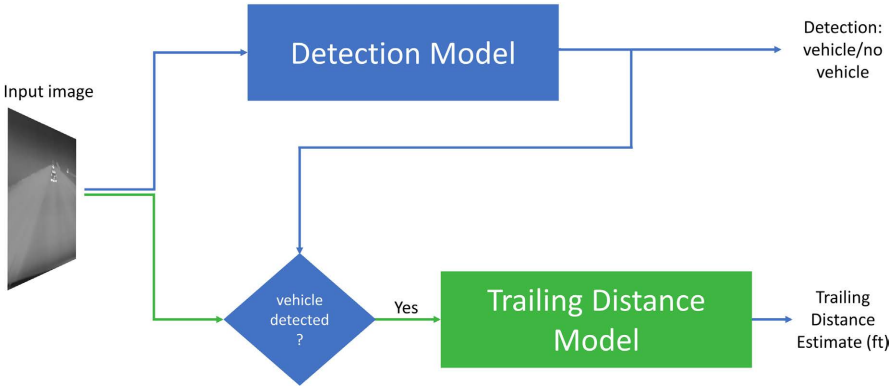


Figure 4. Diagram of inference execution for detection and trailing distance models.

detect the presence or absence of a trailing vehicle in the preprocessed thermal image data. If the detection model detects the presence of a trailing vehicle with a high confidence level, then the Trailing Distance Model (TDM) predicts the trailing distance using the same preprocessed thermal image data. Using this cascade approach, we reduce the computational burden and, in turn, the energy consumption of an edge AI device hosting the two models mounted in the snowplow.

4.1. Detection

The Detection Model is envisioned to perform binary classification (presence or absence of the trailing vehicle) of the preprocessed thermal image and output the probability (confidence level) of the thermal image containing a trailing vehicle. We developed the first detection model consisting of an eighteen-layer deep conventional convolution neural network (CNN) and evaluated its performance. The architectural details, training and validation loss/accuracy plots, and the confusion matrix of the model are presented in the appendix. The model was trained for 100 epochs and achieved a training and validation accuracy of 70.0% and 70.8%, respectively. However, the trained inference performance was poor, as the test accuracy was only 71.0%. The factors contributing to the model's low inference performance are discussed below:

- The network, having eighteen convolution layers and a fully connected layer, exhibited overfitting as the dataset consists of only 16,081 training data. The overfitting can be observed in **Figure A1** and **Figure A2** (see Appendix A1depicting the training and validation loss/accuracy plots.
- The model exhibited memorization of images with vehicles, which can be seen in the confusion matrix of the test dataset in **Figure A3**. In the confusion matrix, it can be seen that the trained model has classified 594 test images with no vehicles as images with vehicles.
- Due to max-pooling layers, the input image was downsized to 2×2 size after flowing through the eighteen convolution layers. As discussed in the section 0.0.3, trailing vehicles can be far from the thermal camera, and objects of interest may be represented only by a few pixels. The downsizing due to using

max-pooling layers will result in the loss of object information.

- The loss during the training remains constant after 40 epochs shown in **Figure A1** indicating a vanishing gradient problem, hindering the network from learning.

To improve the performance of the first prototype detection model, we applied several regularization techniques such as batch normalization, L2 regularization, and data augmentation without modifying the network architecture and training for 100 epochs. The performance of the first prototype detection model with regularization improved significantly, as achieved training and validation accuracy of 99.6% and 99.4%, respectively. The model inference performance with the test data also increased to 94%. Even though the loss for both training and validation are decreasing, as shown in **Figure A4**, the decrease of the loss after 60 epochs is small, and the validation loss is noisy and diverging from the training loss. This loss behavior of the model can still be due to the vanishing gradient problem. Furthermore, the inference performance is not at the expected level shown in the confusion matrix of the test dataset in **Figure A6**. In the confusion matrix, it can be observed that even with regularization techniques, a significant number of non-vehicle thermal images are classified as thermal images with vehicles. Therefore, the model architecture of the first prototype detection model has a low generalization capability.

To improve the generalization capability and address the overfitting and vanishing gradient issues, we propose using Residual Networks (ResNets), a deep learning architecture which is a reformulated Convolutional Neural Network (CNN). ResNets are a reformulation of the classical CNN networks, where, in addition to feeding the output feature map of one convolution layer as input of the next layer, the layers also contain shortcut connections, adding its input feature map to the output of the same layer. The shortcut connections to a layer subject the layer to its own convolution operation, which serves to match the dimensionality input feature map to the output map to which it is added. The shortcut connections are conceptualized as a *residual block*, as shown in **Figure 5**.

Residual networks have several advantages over the classical CNN networks [16]:

- The skip connections, allowing gradients to flow to previous layers during backpropagation without being attenuated, addresses the vanishing gradient problem.
- ResNets reduce network degradation for deeper networks by reformulating the desired mapping $H(x)$ to a residual mapping $F(x)=H(x)-x$ and recasting the original mapping to $F(x)+x$. While theoretically, a series of non-linear layers should be able to approximate either mapping, ResNets are empirically shown to improve optimization.
- By providing multiple paths for data to flow through the network, ResNets implement an inherent form of regularization. Similar to Dropout [17], skip connections provide redundancy in propagating important features through the network.

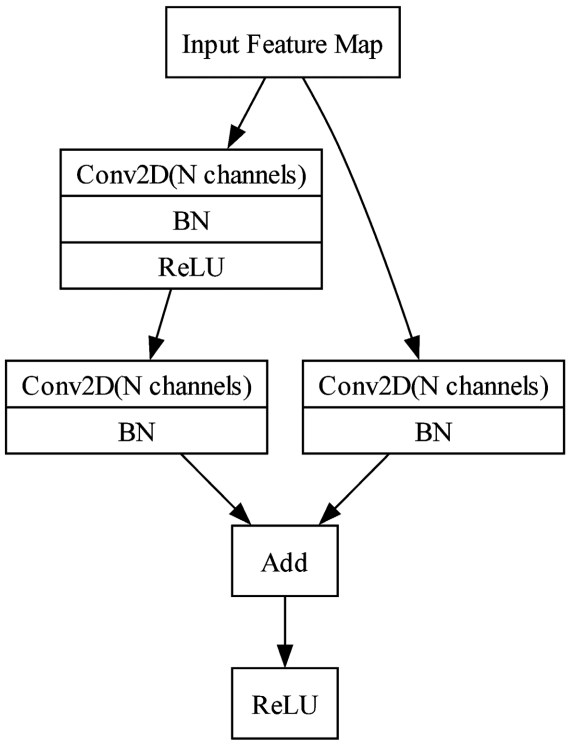


Figure 5. Residual convolution block.

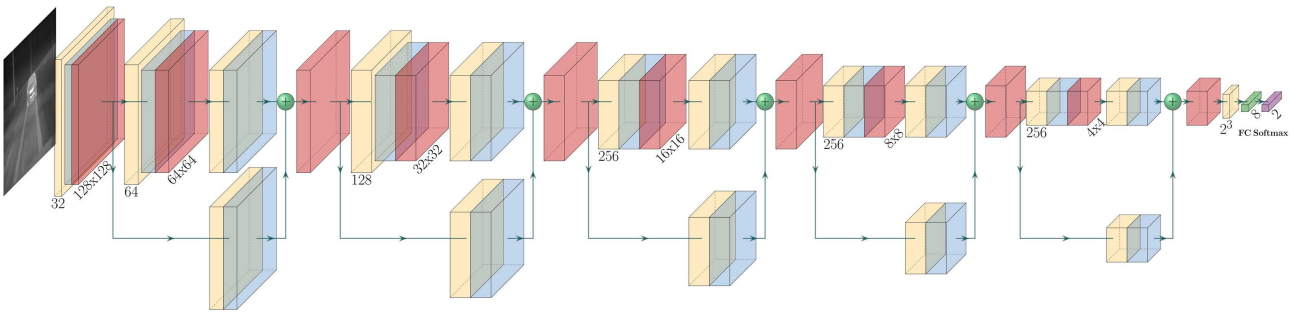


Figure 6. ResNet based detection model architecture.

The Resnet-based detection network architecture is shown in **Figure 6**. The network consists of Conv2D, Batch Normalization, and fully connected layers. In **Figure 6**, the yellow-colored blocks represent the Conv2D layers, the batch normalization layers are represented by blue-colored blocks, followed by the ReLU activation blocks in red-color. After the residual blocks, the network has a fully connected layer represented by the green-colored block, followed by a softmax layer with two outputs.

The original residual network [16] proposed by Kaiming et. al., do not implement pooling layers in the residual blocks, but do use pooling layers at the beginning and end of the network to downsize the feature maps to the desired dimension for the following layers. In our proposed ResNet, the max and average pooling operations are eliminated completely to avoid losing individual pixel information. The discarding of individual pixel information is inherent due to

pooling layers and is undesirable, due to a trailing vehicle being represented by a few pixels at large distances. Due to the vehicle constituent pixel area being small, the use of pooling layers will eventually result in the vehicle segment being represented by a single pixel in an input feature map. Convolution layers are trainable and thus can learn operations that can propagate more information than pooling [18]. Thus, convolution layers with a stride greater than one are used to reduce the spatial dimensions (height and width) of the feature maps.

Additionally, the original ResNet [16] contained two types of residual blocks, characterized by whether they are *bottleneck* blocks or not. Bottleneck blocks are residual blocks in which the first convolution layer has a stride greater than 1, and thus, the feature maps' spatial dimensions are reduced from input to output. Bottleneck blocks also increase the number of feature maps by increasing the number of kernels in the second convolution layer to accompany the reduced spatial dimension. In order for the input to be the same shape as the feature map to which it is added via its skip connection, a convolution layer with 1×1 kernels is used to increase the feature map depth appropriately.

In **Figure 6**, it can be seen that the architecture has only five residual layers, all of which are bottleneck layers according to [16]. This makes the detection model based on the ResNet architecture a rather shallow network by the modern deep convolutional network standards. Therefore, the network architecture can be qualified more specifically as a wide ResNet, which has exhibited superior generalization than deeper ResNets [19].

4.2. Trailing Distance Model

The Trailing Distance Model output is a prediction of the distance between the snowplow and a following vehicle. Therefore, the model has to be designed as a univariate regression problem. We propose another ResNet network with an architecture similar to the detection network, having an image as input and producing a non-negative real number output, which is an estimate of the nearest trailing vehicle distance from the thermal camera mounted at the rear of the snowplow.

There are existing robust approaches of regression using CNN networks to estimate multiple vehicles' distances with no geometric calibrations or conditioning. These robust approaches require large datasets for training. In this work, creating a large thermal image dataset with distance as labels is difficult. Collecting a large volume of thermal image data in heavy blowing snow conditions with corresponding distance labels is hindered due to various insurmountable constraints, such as inaccurate measurements by LiDAR in heavy snow, due to high reflectivity and attenuation of light pulses [6], and a radar lacking the ability to distinguish between a vehicle and a background or foreground object due to resolution. Using our GPS/LoRa system to collect distances along with image data, we are limited to a single-follower vehicle in a frame having a distance measurement. Therefore, our current training dataset has images with the fol-

lower vehicle in front of all other vehicles on the road, such that distance labels correspond to the geometry of the image scenes.

The primary image feature in this regression application is the number of pixels that represent the trailing vehicle. Thus, each reduction in the dimension of the feature maps destroys the pixel area information, reducing the resolution, which can be catastrophic to the network's ability to predict distance accurately. Choosing a network with less number of residual blocks results in larger feature maps as input to the output layer, avoiding the destruction of the pixel area information. The number of residual blocks in the ResNet architecture for the TDM network is decreased by one, and this reduction was noticed to reduce the variance of target prediction error, resulting in a shallower network compared to the Detection network as shown in Figure 7. Furthermore, due to the limited data set, avoiding overfitting is essential to reduce the variance of prediction error, especially as vehicles become more similar in appearance at large distances. Thus, we choose global average pooling (GAP) to transform the larger feature maps into a vector for the fully connected layers without any trainable parameters. In Figure 7, the purple-colored block represents the GAP.

5. Network Training

As discussed previously, in section 3.1, a data set of thermal images of trailing vehicles was created during several heavy snow events on two-lane highways and interstate roads. During the dataset creation drives, using the LoRa/GPA device, the trailing distance was collected for each image frame. Later, the images in the dataset were manually labeled, identifying the presence or absence of the trailing vehicle.

Given a training set of input-output pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$ and a neural network which maps inputs to predicted outputs as $\hat{\mathbf{y}}_i = f(\mathbf{x}_i; \mathbf{w})$ with parameters $\mathbf{w}$. We seek to design the network parameters $\mathbf{w}$ to map the predicted outputs $\hat{\mathbf{y}}_i$ close to the ground truth $\mathbf{y}_i$ for each input $\mathbf{x}_i$. However, due to the large number of parameters, even in small neural networks, it is not possible to manually design the network parameters to fit a particular dataset. To address this, a task-dependent *loss function* $L\left[f(\mathbf{x}_i; \mathbf{w}), \mathbf{y}_i\right]$ shown in Equation (4) is defined to quantify the mismatch between predictions and labels (the ground truth) for each sample. Thus, the problem of network fitting is framed as an

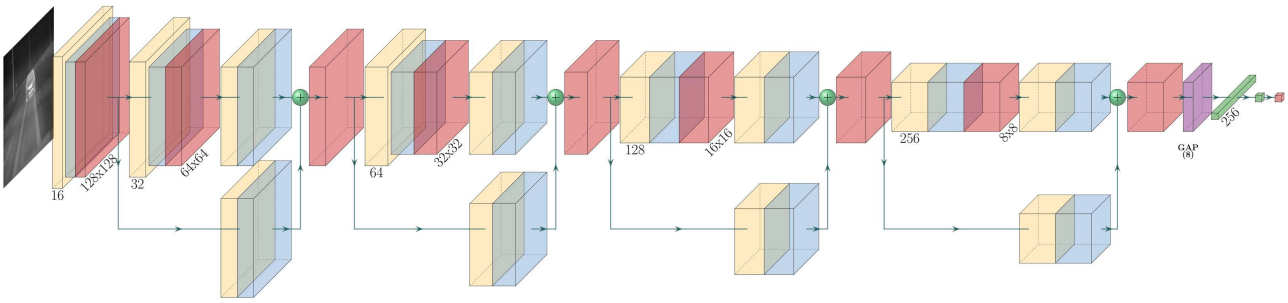


Figure 7. ResNet based trailing distance model architecture.

optimization problem, with an optimization algorithm seeking to minimize the loss by updating the network parameters iteratively and finding an optimal parameter set.

$$\hat{\mathbf{w}} = \arg\min_{w} L\left[f\left(\mathbf{x}_i; \mathbf{w}\right), \mathbf{y}_i \right]. \tag{4}$$

The fundamental optimization technique for training neural networks is the Stochastic Gradient Descent (SGD) algorithm. In SGD, the network parameters are initialized with small random values and then updated repeatedly by presenting mini-batches of input-label pairs to the network. These mini-batches of data are randomly sampled from the training set, allowing the optimization to take small sub-optimal steps, *i.e.*, travel downhill on the loss function surface towards a minimum, which may be local or global. In each iteration, the mini-batch of inputs is passed through the network, obtaining predictions $\hat{\mathbf{y}}_i = f\left(\mathbf{x}_i; \mathbf{w}\right)$. Then, the loss is calculated per sample using each sample's corresponding label $\mathbf{y}_i$. To perform gradient descent, the derivatives of the loss with respect to the network parameters, known as the gradient of error, are calculated as shown in Equation (5) using a small *learning rate* α, which is one of the hyperparameters.

$$\nabla_{\mathbf{w}} L = \left[\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \cdots, \frac{\partial L}{\partial w_d} \right]^{\mathrm{T}}. \tag{5}$$

After computing the gradient of error, the parameters are adjusted to reduce the loss value as shown in Equation (6):

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla_{\mathbf{w}} L \tag{6}$$

Since the training process is done by presenting mini-batches of samples at each iteration, the parameter update must account for the loss for each sample, which is done by simply accumulating the partial derivatives of the loss as given by the Equation (7).

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \alpha \sum_{i \in B_t} \frac{\partial L_i}{\partial \mathbf{w}} \tag{7}$$

where B_t is the set comprising the mini-batch of input-label pairs and L_i is the loss value for the i^{th} pair. In the case of a neural network that is more than one layer deep, the partial derivatives of the loss w.r.t. the parameters are more complicated, as the network parameters are distributed throughout the loss function. The network parameters are updated layer by layer using the *backpropagation* algorithm. The networks used in this research were implemented and trained using PyTorch, a deep-learning library for the Python programming language.

5.1. Detection Network Training

The data set we cultivated is comprised of 20,107 thermal images, of which approximately 67% contain vehicles. This data set is split into training, validation, and test subsets with ratios of 0.8, 0.1, and 0.1, respectively.

A list of the hyperparameters used for training the ResNet-based detection network is shown in Table 1.

The loss function used with the detection network is the Binary Cross Entropy (BCE) loss shown in Equation (8), which measures the dissimilarity between the predicted class for each sample and the corresponding class label.

$$L_{BCE}\left(\mathbf{y},\hat{\mathbf{y}}\right)=-\frac{1}{N}\sum_{i=1}^{N}y_i\cdot\log\left(\hat{y}_i\right)+\left(1-y_i\right)\cdot\log\left(1-\hat{y}_i\right) \tag{8}$$

where y_i is the true label of the sample i (e.g. frame contains a vehicle), $\hat{y}_i=f\left(x_i\right)$ is the prediction of the network, *i.e.*, the probability of belonging to the positive class. This loss function simultaneously measures the difference between the predicted probabilities to actual class belonging, 0 or 1, of each sample. In minimizing this loss function, predictions that are far from their corresponding label are penalized by accumulating a higher loss value. Since $\hat{y}_i=f\left(x_i\right)$ is differentiable with respect to the network parameters, the BCE loss can be optimized to align the network predictions with the ground truth across the training set. Adaptive Momentum (Adam) is a stochastic optimization algorithm that improves the fundamental SGD algorithm [20]. The SGD algorithm with a fixed step size has undesirable optimization properties, such as large adjustments to parameters corresponding to large gradients that can cause the weight update to overshoot and small adjustments to parameters corresponding to small gradients that can cause slow progress towards nearby minima. These characteristics of SGD make it difficult to reach the minima of the loss surface at a sufficient rate while remaining stable. Adam addresses these concerns first by normalizing the gradients, which makes the weight updates move a distance fixed by the learning rate in all directions. However, a fixed step size would not allow the optimizer to converge, instead oscillating around minima, as well as not allowing the optimizer to take larger or smaller steps when necessary. To address this, Adam adds momentum to the estimate [20].

The step decay learning rate schedule was chosen to attenuate large oscillations in loss during the later phases of training. During the initial phases of training, the parameters are initialized with random values, resulting in a large

Table 1. Detection network training and optimization details.

Hyperparameters	Details
Loss Function	Binary Cross Entropy Loss
Optimizer	Adam
Initial Learning Rate	1e−3
Learning Rate Scheduler	Geometric Decay, step size = 10, multiplier = 0.85
Minibatch Size	128
Epochs	100
Parameter Regularization	L2 Weight Decay, $\lambda=0.002$

loss, *i.e.*, the initial loss is far from any local or global minimum. Thus, it is valuable to introduce a mechanism that encourages large steps at the beginning of training to explore the loss surface sufficiently and which reduces the optimization step size in later training steps to promote convergence rather than oscillation around a minimum. There are several approaches to learning rate scheduling, but theoretical guarantees of their relative efficacy cannot be made due to the high dimensionality of the loss surface with a deep neural network architecture. Thus, we selected the Geometric learning rate decay, otherwise known as a step decay schedule a learning rate schedule that reduces training over time and is simple and intuitive to tune. Furthermore, the Geometric learning rate decay reduces the learning rate by a multiplicative factor in each epoch, as given by Equation (9).

$$\lambda_t = \lambda_0 \times \gamma^{\lfloor t/s \rfloor} \tag{9}$$

where λ is the learning rate at epoch t, λ_0 is the initial learning rate, s is the step size, and $\lfloor . \rfloor$ indicates the floor operation. Figure 8 depicts the geometric learning decay using the multiplicative factor and step size parameters in Table 1.

In addition to the use of batch normalization and skip connections, L2 weight

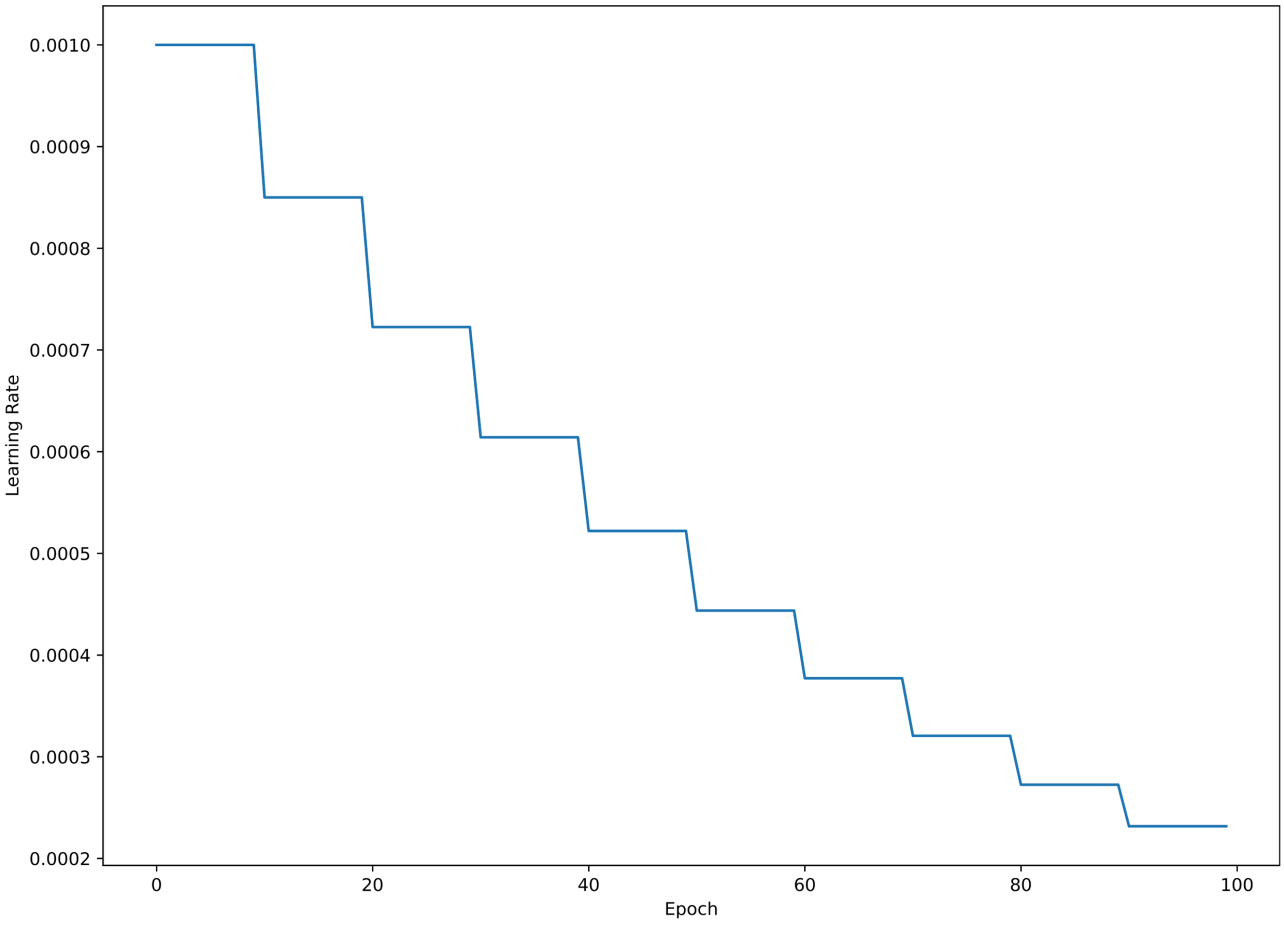


Figure 8. Geometric decay learning rate scheduler.

decay is also applied to encourage smoother change in weights and thus achieve a better conditioned objective surface. Additionally, L2 regularization is used to combat the overfitting problem due to the limited thermal image dataset. Since the thermal images can be collected only during severe weather events, the number of thermal images and the variety of the images with respect to the severity of snow, types of consumer vehicles, and distance between the snowplow and consumer vehicles is limited.

The network was trained for 100 epochs, and the training and validation loss and accuracy shown in **Figure 9** and **Figure 10** were collected. The regularization techniques used with the first prototype network, the L2 weight decay, batch normalization, and heavy data augmentation, were used in training the ResNet-based Detection network. In **Figure 9**, it can be observed that the validation loss is less than the training loss, which is due to the use of regularization only on the training dataset. Since regularization techniques are not applied during inference with the validation dataset, the network can perform better on the validation dataset. Furthermore, regularization trades the model's fit to the training data for better generalization of unseen data (validation dataset).

Comparing the accuracy plot of ResNet-based Detection network in **Figure 10**

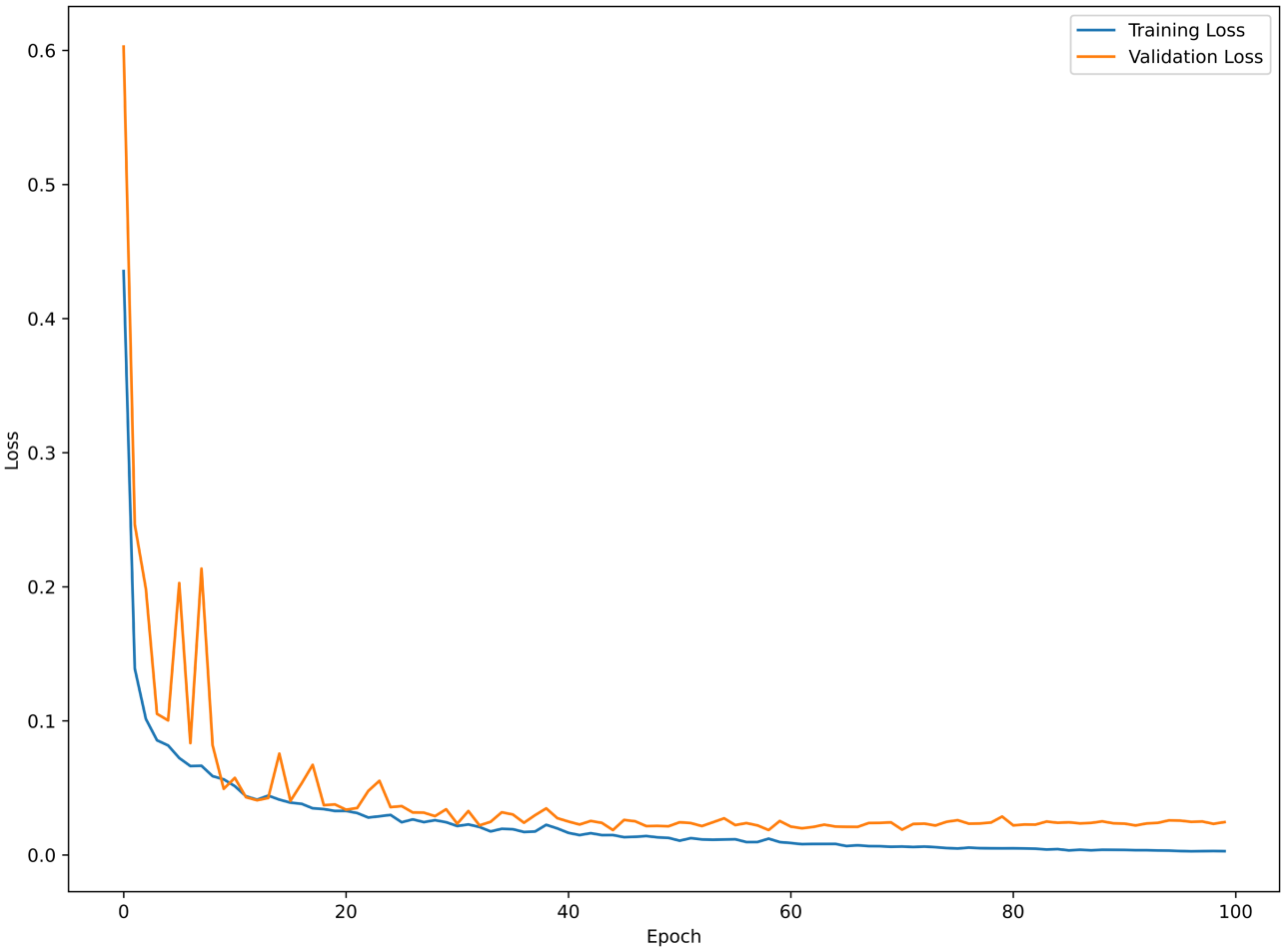


Figure 9. Training and validation loss of ResNet based detection network.

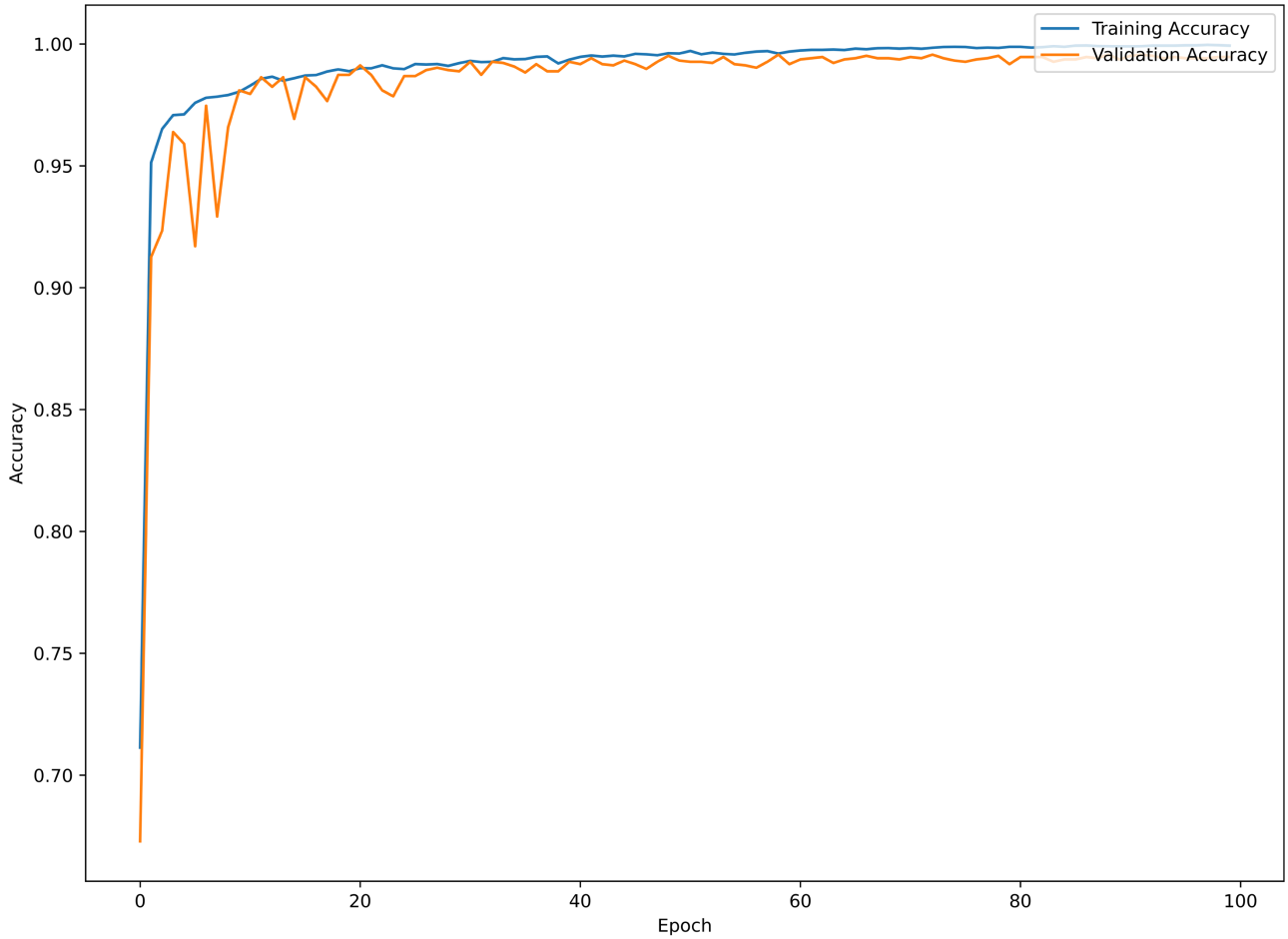


Figure 10. Training and validation accuracy of detection network.

with the accuracy plots of the first prototype network with and without regularization in **Figure A2** & **Figure A5**, it can be observed, a better performance by the ResNet-based Detection network. The ResNet-based Detection network has training and validation accuracy close to 99.99% with less variance. The first prototype network without regularization only had an accuracy of 70.0%, and with regularization, an accuracy of 99.99% with a large variance. Similar performance improvement in the training and validation loss of the ResNet-based Detection network in comparison to the first prototype network can be observed in **Figure 9**, **Figure A1** & **Figure A4**.

5.2. Trailing Distance Network (TDN)

Ideally, each image frame should have only the test trailing vehicle, without other vehicles between the camera on the vehicle simulating a snowplow and trailing vehicle, and with a valid distance label. However, this is not always true, particularly at larger distances, since other consumer vehicles on the road often pass the test trailing vehicle, occluding it from the camera and rendering the LoRa/GPS-based computed distance label invalid. As a result of these label uncertainties, many images used to train the Detection network cannot be used to

train the TDN. Therefore, the dataset used for the training of the TDN is a subset of the collected dataset. The dataset subset is created by selecting images that have only the test trailing vehicle (the positive class of the full dataset), and no other consumer vehicles between the camera and the test trailing vehicle. The TDN dataset consists of 10,460 thermal images, split into 80% training data and 10% validation and test data. Each image in the dataset has a corresponding distance label measured in feet. Figure 11 shows the distribution of distance labels in the training dataset, with very similar distributions for the validation and test set distance labels.

In Figure 11, it can be seen that we have few images at distances greater than 1000 ft. due to other consumer vehicles interfering during the data collection activity. The TDN dataset is skewed towards shorter trailing distances. The hyperparameters used in training the TDN are presented in Table 2.

In contrast to the Detection Network, the Trailing Distance Network must predict a single real value representing the predicted distance of the trailing vehicle, and in training the network, this predicted value must be compared with the true distance label. Therefore, the Mean absolute error (MAE) given in

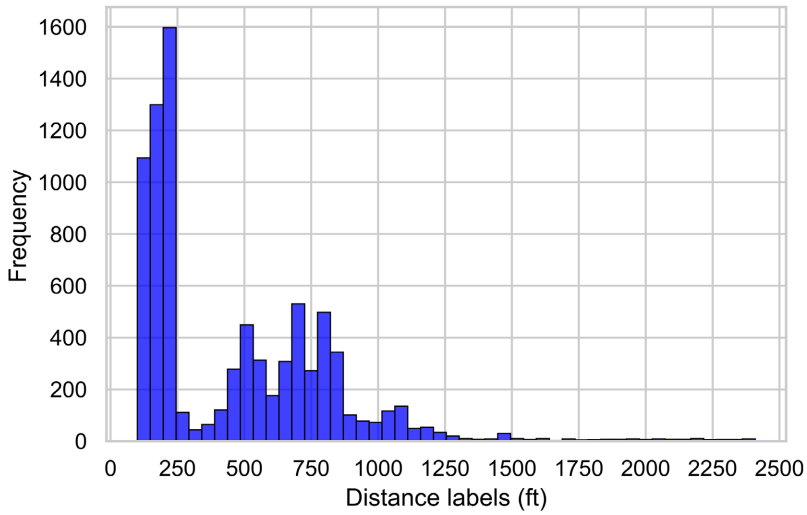


Figure 11. Distribution of distance labels in training data set.

Table 2. Trailing distance network training and optimization details.

Hyperparam. or Reg. Technique	Details
Loss Function	Mean Absolute Error (MAE)
Optimizer	Adam
Initial Learning Rate	1e−3
Learning Rate Scheduler	Geometric Decay, step size = 10, multiplier = 0.825
Minibatch Size	128
Epochs	100
Parameter Regularization	L2 Weight Decay, $\lambda = 0.005$

Equation (10) is a loss function suitable for regression tasks.

$$L_{MAE}(\mathbf{y},\hat{\mathbf{y}}) = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{10}$$

In comparison with other regression loss functions, such as Mean Squared Error loss (MSE), MAE has the advantage of being interpretable in the units of the regression target value, which is distance in this application. Additionally, MAE is less sensitive to outliers than MSE, as MSE squares the difference between predicted value and label, and large differences are more heavily penalized. Due to issues of data integrity and limited availability of large trailing vehicle distances, which will be discussed later, our experimentation showed MAE to exhibit superior performance than MSE during this development phase of the TDN. The hyperparameters of the TDN are similar to that of the Detection network with one difference, *i.e.*, the TDN uses a larger L2 weight penalty to encourage a more regularized model.

Figure 12 shows the MAE loss on training and validation sets per epoch. We

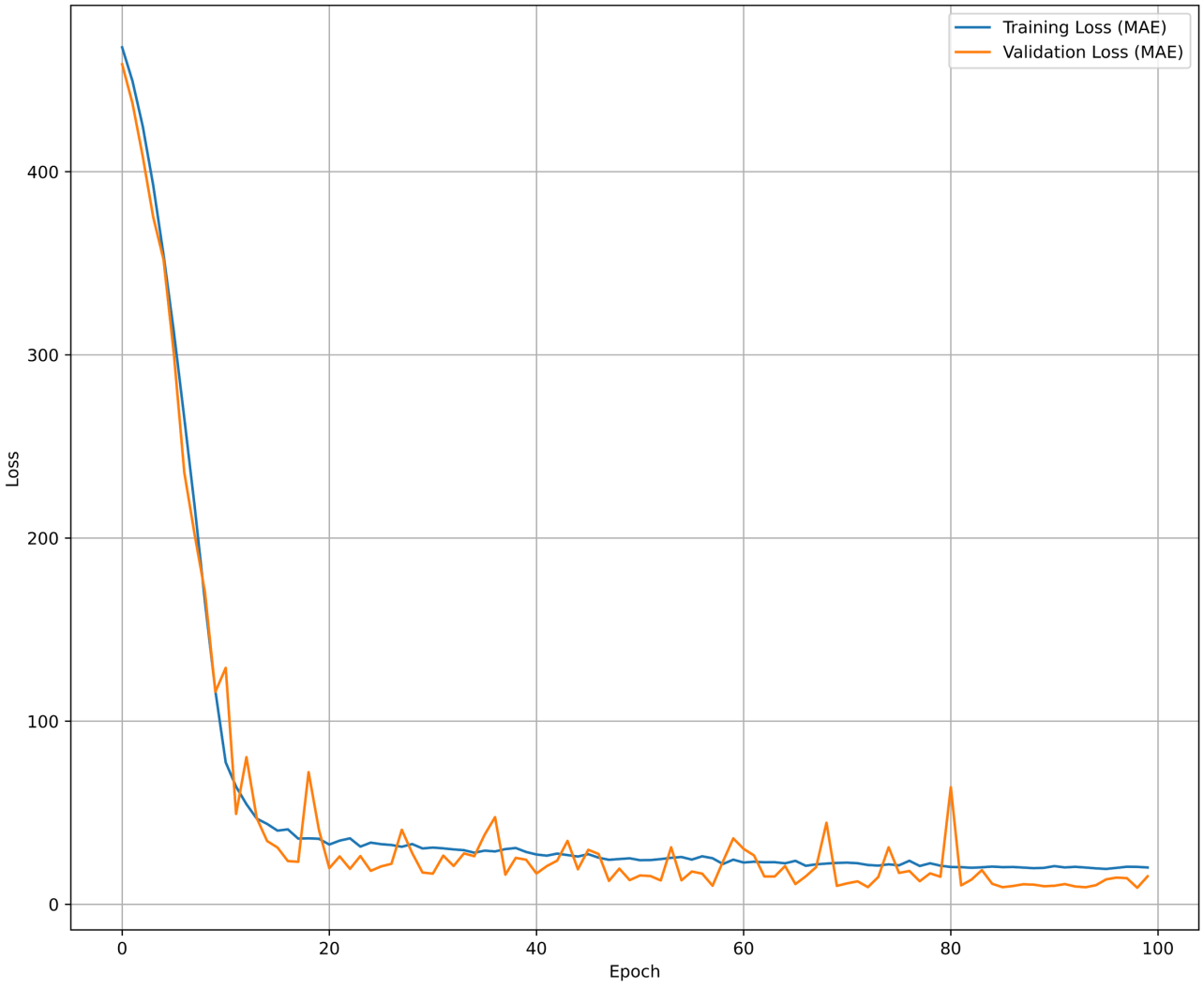


Figure 12. Training and validation loss for trailing distance network during learning.

can see that there is a sharp decline and saturation of the training loss, indicating that the network architecture is unable to reduce the MAE further. The validation loss follows a similar trajectory, albeit with expected oscillations above and below the training loss inherent in mini-batch Stochastic Gradient Descent (SGD) based training.

6. Results

A detailed analysis of the trained detection and trailing distance network performance with the test data, and possible reasons for the shortcomings of the models are presented in this section.

6.1. Detection Network

The confusion matrix of the detection network with the test dataset containing 2071 thermal images is presented in Figure 13. The training dataset is imbalanced towards the positive class (presence of the trailing vehicle) since it contains approximately 2/3 of images with the trailing vehicle. Therefore, the network may have learned a bias toward predicting the presence of a vehicle during training. However, in the confusion matrix, it can be seen that the network has classified only 6 test images out of 600 total no trailing vehicle images as images

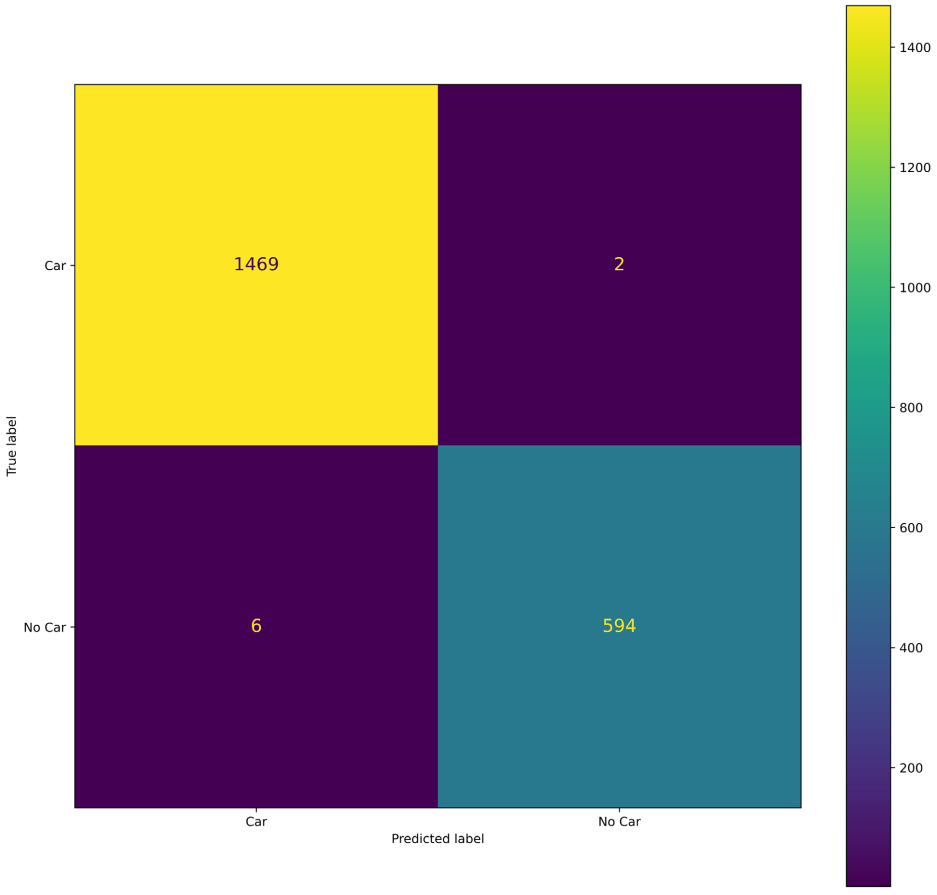


Figure 13. Confusion matrix of trained detection network on test data set.

with trailing vehicle, demonstrating that the trained network is not biased towards the positive class. The network has an overall test accuracy of 99.6%, highlighting that the network has clearly learned several salient features of vehicles in the images and generalization capability. The ResNet-based detection network has predicted only two images with trailing vehicles as images with no trailing vehicles. This is a significant performance improvement compared to the prediction of forty-four images with trailing vehicles as images with no trailing vehicles in the confusion matrix of the classical CNN network in **Figure A6**. It is important to have very low false negatives since missing the presence of a trailing vehicle will prevent the snowplow operator from taking evasive actions, leading to catastrophic accidents.

Using the confusion matrix in **Figure 13**, several key metrics such as precision, recall, and F1-score were computed. The precision metric measures the proportion of positive predictions that are, in fact, positive samples, *i.e.*, the model's accuracy in identifying a vehicle given that the thermal frame contains a vehicle. The high precision value of 0.996 indicates that when the network detects a vehicle, it is highly likely to be an actual vehicle, which minimizes false alarms. In contrast, the recall metric represents the proportion of true positive samples that the model correctly identifies. In this application, this indicates the model's ability to detect vehicles when they are present in the thermal frame. A High recall metric value of 0.999 indicates that the model will be unlikely to miss vehicles when they are present, a crucial aspect of the performance of the model as the most severe outcome of false negatives is a rear-end collision, whereas the most severe outcome of false positives is unnecessary harm reduction actions by the operator. Additionally, the F1-score of 0.991 provides a balance between recall and precision. With the low number of misclassifications, false positives, or false negatives across the test set, all three of these metrics are exemplary, with a slight bias towards recall over precision, which is advantageous in this application, as false negatives can result in rear-end collisions.

We analyze the network with test samples that are classified incorrectly to interpret and explain the network performance, addressing what the network has learned and/or has failed to learn. **Figure 14** depicts two true positive test thermal images, *i.e.*, images that contain a vehicle, but the network predicts them not to contain a vehicle, *i.e.*, false negatives. In both test images, the vehicles are located at the edge of the frame, and only a part of the trailing vehicles are present in the image. The misclassification can be due to two reasons

- The training dataset does not have a significant number of images, with only parts of trailing vehicles visible and labeled as positive. The majority of the images have the trailing vehicles fully represented, and located closer to the center of the image. Hence, the network was not trained to recognize images with parts of trailing vehicles located closer to the edge of an image. This is due to the data collection occurring mainly on straight roadways with a clear line of sight between the camera and trailing vehicles.

- As these kinds of images flow through the multiple convolution layers, the output feature maps of each layer could have only identified the edges of an image instead of the features of a partial vehicle.

Figure 15 shows a cause of false positive predictions in low-resolution infrared imaging with interfering objects of intense heat signatures. Ideally, the thermal camera is expected to capture the trailing vehicles as having the most intense heat signatures in snowy road conditions. However, Figure 15 has an industrial gravel plant in the background, which is a significant heat source and exudes intense infrared radiation. The red arrow in Figure 15 points to a part of the trailing vehicle entering the frame. In our labeling procedure, we have labeled images with only slivers of trailing vehicles as the negative class to reduce the probability of the network learning random structures as vehicles. However, the proximity of the industrial site with intense infrared radiation signature at the location where thermal image data was collected has caused edge cases like this to exist in our training data and confused the network during training.

Another false positive sample is shown in Figure 16. In this image, a vehicle can be seen clearly by the naked eye. However, it doesn't contain the test trailing vehicle, as the vehicle is a van, is off the road, and is not traveling in the same

Figure 14. Images misclassified as not containing a trailing vehicle.

Figure 15. Image misclassified as containing a trailing vehicle.

direction. Our labeling criteria were to include only trailing vehicles in the positive class. Thus, even though the network successfully classified the presence of a vehicle, this prediction did not align with the chosen label. This may be an instance of erroneous data labeling; our approach to labeling may need refinement.

The other false positive sample of note is shown in Figure 17. Here, the trailing vehicle was just over 1000 ft away from the thermal camera, and thus, the features of the trailing vehicle are small and difficult to learn and represent in the feature maps. As a result, this image was labeled incorrectly as not containing a trailing vehicle, and therefore, the prediction as a positive class is actually correct. This highlights an important issue with data labeling. Since vehicles at far distances are represented only by a few pixels, the low resolution, dilated appearance, and lack of distinguishing color information, such samples can be visually misinterpreted. Since this is a large bespoke data set that we collected and labeled manually, mistakes such as this are possible.

Figure 16. Image incorrectly predicted as containing a trailing vehicle.

Figure 17. Image incorrectly predicted as containing a trailing vehicle.

6.2. Trailing Distance Network

After training, the test dataset consisting of 1046 thermal images labeled with the measured distances was presented to the network for inference. The network achieved an MAE of 10.70 ft. with a standard deviation of 14.01 ft. Since the label and prediction are both non-negative real numbers, the MAE represents the average distance in feet that the network prediction varies from the true distance. An MAE of 10.7 ft with a standard deviation of 14.01 ft can be categorized as a highly accurate prediction in the context of the snowplow application. Since a snowplow operator is expected to take evasive actions when the trailing vehicle is approximately at a distance of 500 ft., an MAE of 10.70 ft. is acceptable for the actual trailing distances ranging from 150 to 2300 ft., as shown in Figure 11.

To analyze further, a violin plot of the distance estimation error of the test dataset is presented in Figure 18. In the violin plot, samples are grouped into 100 ft bins according to their true distance labels. The distribution of the distance estimation error of each distance bin is illustrated via the thickness and distance of lines across the vertical axis.

From Figure 18, it is clear that the mean prediction error increases as the trailing distance of the vehicle is large. Moreover, the deviation of the prediction error is also large at greater distances. There are three potential reasons for this behavior, all of which likely contribute to the positive correlation between vehicle distance and prediction error magnitude:

1) As the trailing vehicle distance from the camera is large, its constituent

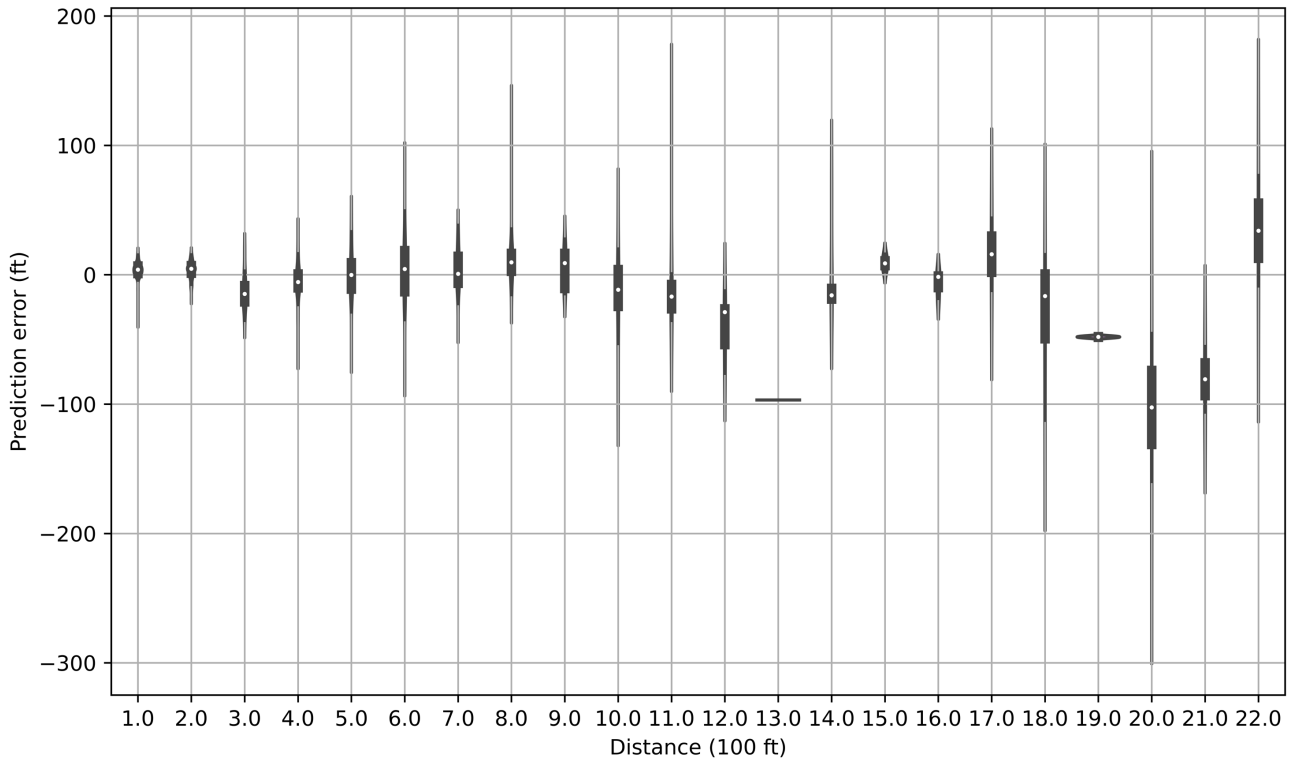


Figure 18. Violin plot of the distance estimation error.

pixel area is smaller. Additionally, at far distances, small differences in distance correspond to very subtle changes in the pixel area which constitutes the trailing vehicle. Since convolution layers degrade pixel information when downsizing their feature maps, these subtle distinctions are lost.

2) As shown in Figure 11, the dataset is heavily skewed towards images at short trailing vehicle distances labels, *i.e.* the bulk of the training samples lie within 100 - 800 ft. As a result, the TDN was trained with a few images of the trailing vehicle at large distances and thereby has not learned the mapping of small pixel areas found at large distances.

3) Despite our best efforts, the data collection process (see 3.1) is not impervious to measurement error. As the distance between the lead and the trailing vehicle increases, especially during large snow events with high moisture and atmospheric snow volume, the LoRa communication protocol for exchanging GPS data has been observed to lose connection intermittently. Thus, as distance increases in adverse weather conditions, the accuracy of the distance labels decreases.

6.3. Implementation

There are two key issues concerning the practical implementation of the Detection and Trailing Distance Estimation models on a snowplow: whether the network predictions can generalize to real-time thermal data and engineering the system to maintain data integrity. To assess the performance of the models with data collected outside of the training and test sets, a field test of the system was performed.

The field test was performed as follows:

1) On a long flat road; the instruments were mounted facing rearward on a passenger vehicle (leader).

2) At an initial distance of 1000 ft behind the leader, another passenger vehicle (follower) was stationary on the road in view of the thermal and RGB cameras.

3) At 1000 ft, real-time inference was performed on collected thermal images containing the follower for both the trained DN and TDN.

4) The follower approached the leader in increments of 100 ft, measured using a handheld laser range finder, and remained stationary at these distances for another cycle of collection and inference.

5) Inferences were collected at 100 ft increments beginning at 1000 ft and ending at 100 ft.

Sample images at 1000, 500, and 100 feet are shown in Figure 19 through Figure 21.

With all field test images containing a vehicle, the Detection Network correctly predicted the presence of a trailing vehicle in all tested thermal samples. In the laboratory setup, the detection network was tested with images not having vehicles, and it accurately predicted the lack of a vehicle in images. Figure 22 demonstrates the error of the Trailing Distance Network predictions:

Figure 19. Thermal and corresponding RGB images of field test at 1000 ft.

Figure 20. Thermal and corresponding RGB images of field test at 500 ft.

Figure 21. Thermal and corresponding RGb images of field test at 100 ft.

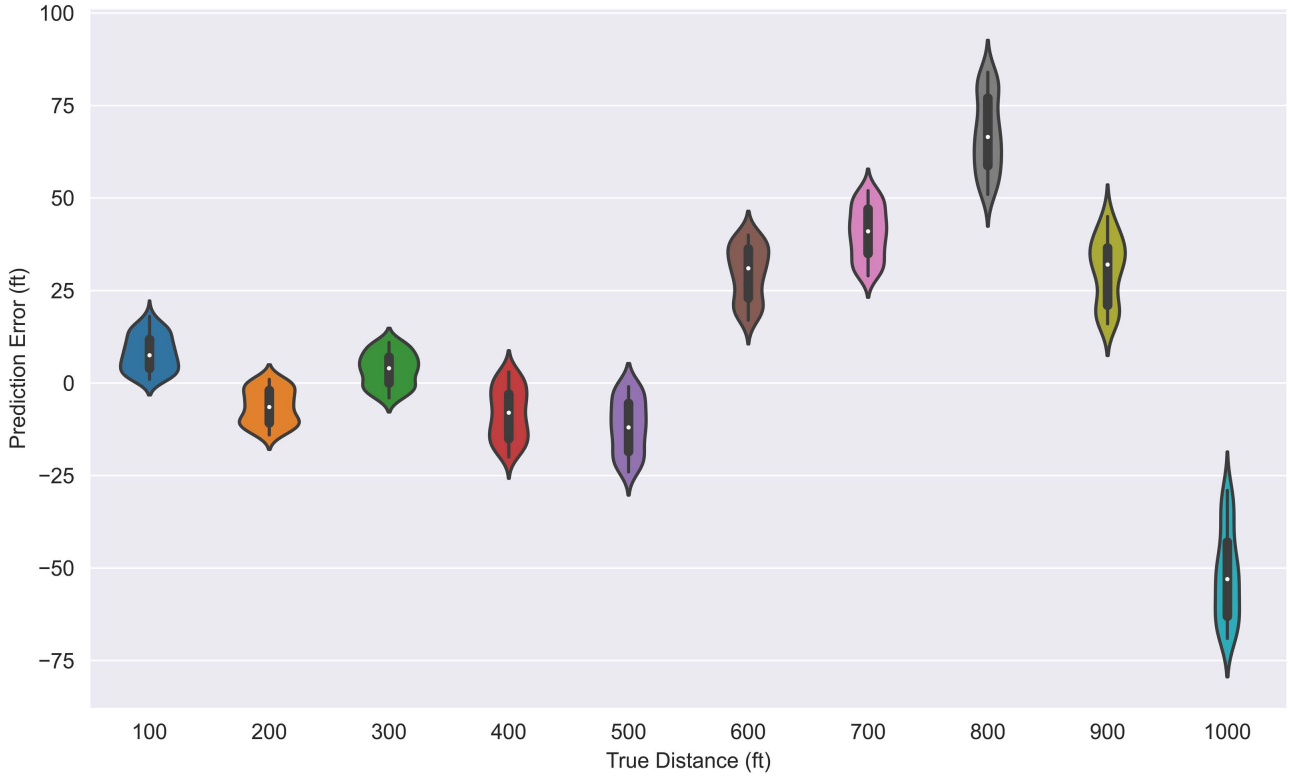


Figure 22. Violin plot of distance estimation error with field test data.

Figure 22 shows similar results, which align with our analysis of the limitations of monocular distance estimation using pixel area. At close range, the network predictions have low mean error compared with ground truth distance measurements. However, as trailing vehicle distance increases, there is a subsequent increase in the distance estimation by the TDN, with both mean error and error deviation increasing. As mentioned previously, this is likely due to the smaller absolute vehicle pixel area at large distances, as well as the smaller differential in pixel area between samples at different distances, which are both large in magnitude.

The other practical obstacle in implementing the collision avoidance system in real-world settings is mounting the snowplow and maintaining its functionality on a snowplow. Due to the large volume of snow displaced by the snowplow blade, snowplows tend to accumulate snow build-up on the rear of the plow. Additionally, many plows release dirt, salt, or other materials as they operate to assist in melting ice on the roads. Thus, the system must have mechanisms in place to remove snow accumulation from the instruments and protect them from damage and wear from dirt and salt splash.

7. Conclusions and Future Work

In this paper, we have detailed the prototype development of an early-collision warning system for DOT snow plows. The main sensor of this prototype is an infrared camera, which we used to cultivate a data set of thermal images with corresponding vehicle distances. A custom thermal image preprocessing sequence was developed and envisioned to increase the saliency of key features in the thermal images. We first developed a deep classical convolutional neural network to perform vehicle detection and identified the issues of overfitting and vanishing gradient contributing to low performance. We designed detection and trailing distance networks based on the modified ResNet architecture. We demonstrated that the residual connections improve the learning of salient features by allowing proper gradient flow during training. We then observed excellent test accuracy for the detection model, as well as the rationale for why certain samples were misclassified. Finally, we showed promising results for trailing distance estimation. Overall, low MAE on the test set indicates a good starting point for distance estimation. However, a large variance in distance estimation indicates a need for improvement of the trailing distance network.

As with all practical applications of deep learning, the fundamental limiting factors in performance remain the size, quality, and variability of the data set. In this setting, the data set appears to be insufficient to train a distance estimation model to predict vehicle distance consistently over a large range. We propose that the TDN performance can be improved by one or more of the following approaches:

- **Large Thermal Image Dataset Creation**

  Collecting more data using the same process as in section 0 during future

snow events will allow representation of new types of trailing vehicles, a larger variety of environmental features, and an improvement of the balance of the data set with respect to their distance labels. However, driving in heavy snow conditions is costly and dangerous.

- **Augmentation of the Thermal Image Dataset with Synthetic Images**

  Research is ongoing into supplementing our training data sets with synthetically generated data. Using state-of-the-art deep generative models, such as StyleGAN and Latent Diffusion, the existing thermal image set can be used to train these models to generate synthetic thermal images which can be used to train both Detection and Trailing Distance networks. However, this does not address the data imbalance problem, as generative models will only sample from their training data distribution.

- **Exploration of Network Architectures for Distance Estimation of Small Objects**

  Vehicle distance is intrinsically correlated with pixel area, and it is likely that continuously downsizing the feature maps due to successive convolution layers with stride is degrading information regarding pixel area in the CNN architecture used for the TDN. It is conceivable that an architecture that reduces the feature maps in some other way can be leveraged to improve regression at large distances. While small object detection is an open research question, there are at least two possible directions that could be explored.

  - **Wide Convolutional Architecture:**

  A network with a large number of feature maps in the initial layer and a reduced number of sequential convolution layers, thus wide rather than deep, reduces the deleterious effect of subsequent downsizing operations on pixel area. However, wide and shallow networks are often not very expressive due to their lack of composition.

  - **Object Detection:**

  The second approach would be to use an object detection approach, which would localize vehicles and predict a bounding box around the pixels corresponding to the trailing vehicle. The bounding box dimensions could ostensibly be used to assist in estimating the trailing distance. There are many disparate approaches to small object detection [21], although none stand out as a clearly superior proposition with a low-resolution thermal data set. This approach would also necessitate more detailed annotations to the training set.

Acknowledgements

We express our sincere thanks to the Wyoming Department of Transportation (WYDOT) for providing grant funding to conduct this research work.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] (2021) Snowplow Collisions. https://www.dot.state.wy.us/news/multiple-snowplows-hit-over-a-five-day-period/

[2] Zockaie, A., Saedi, R., Gates, T.J., Savolainen, P.T., Schneider, B., Ghamami, M., Verma, R., Fakhrmoosavi, F., Kavianipour, M., Shojaei, M., *et al.* (2018) Evaluation of a Collision Avoidance and Mitigation System (CAMS) on Winter Maintenance Trucks. Technical Report, Michigan.

[3] Camden, M.C., Hickman, J.S., Tidwell, S., Soccolich, S.A., Hammond, R., Hanowski, R.J., et al. (2020) Defensive Driving for Snowplow Operators. Technical Report, Minnesota. Department of Transportation. Clear Roads Pooled Fund.

[4] Haq, M.T., Reza, I. and Ksaibati, K. (2023) Investigating Snowplow-Related Injury Severity along Mountainous Roadway in Wyoming. *Journal of Sustainable deVelopment of Transport and Logistics*, **8**, 73-88. https://doi.org/10.14254/jsdtl.2023.8-1.6

[5] Warren, S.G. (2019) Optical Properties of Ice and Snow. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **377**, Article ID: 20180161. https://doi.org/10.1098/rsta.2018.0161

[6] Nebuloni, R. and Capsoni, C. (2008) Laser Attenuation by Falling Snow. 2008 6*th International Symposium on Communication Systems, Networks and Digital Signal Processing*, Graz, 25 July 2008, 265-269. https://doi.org/10.1109/CSNDSP.2008.4610768

[7] Siddiqui, M.Q. and Ashour, M.W. (2021) Object/Obstacles Detection System for Self-Driving Cars. 4*th Smart Cities Symposium* (*SCS* 2021), 21-23 November 2021, 164-169. https://doi.org/10.1049/icp.2022.0333

[8] Bhadoriya, A.S., Vegamoor, V. and Rathinam, S. (2022) Vehicle Detection and Tracking Using Thermal Cameras in Adverse Visibility Conditions. *Sensors*, 22, Article 4567. https://doi.org/10.3390/s22124567

[9] Alhammadi, S.A., Alhameli, S.A., Almaazmi, F.A., Almazrouei, B.H., Almessabi, H.A. and Abu-Kheil, Y. (2022) Thermal-Based Vehicle Detection System Using Deep Transfer Learning under Extreme Weather Conditions. 2022 8*th International Conference on Information Technology Trends* (*ITT*), Dubai, 25-26 May 2022, 119-123. https://doi.org/10.1109/ITT56123.2022.9863963

[10] Lu, Y.F., Yang, Q.F., Han, J.X. and Zheng, C.H. (2021) A Robust Vehicle Detection Method in Thermal Images Based on Deep Learning. 2021 *IEEE International Conference on Power, Intelligent Computing and Systems* (*ICPICS*), Shenyang, 29-31 July 2021, 386-390. https://doi.org/10.1109/ICPICS52425.2021.9524280

[11] Kang, Q., Zhao, H.D., Yang, D.X., Ahmed, H.S. and Ma, J.C. (2020) Lightweight Convolutional Neural Network for Vehicle Recognition in Thermal Infrared Images. *Infrared Physics & Technology*, **104**, Article ID: 103120. https://doi.org/10.1016/j.infrared.2019.103120

[12] Han, Y.J. and Hu, D. (2020) Multispectral Fusion Approach for Traffic Target Detection in Bad Weather. *Algorithms*, **13**, Article 271. https://doi.org/10.3390/a13110271

[13] Urone, P.P. and Hinrichs, R. (2022) College Physics 2e. Openstax.

[14] Kennedy, H.V. (1993) Modeling Noise in Thermal Imaging Systems. In: Holst, G.C., Ed., *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing IV*, International Society for Optics and Photonics, Bellingham, 66-77. https://doi.org/10.1117/12.154739

[15] Buades, A., Coll, B. and Morel, J.M. (2011) Non-Local Means Denoising. *Image Processing On Line*, **1**, 208-212. https://doi.org/10.5201/ipol.2011.bcm_nlm

[16] He, K.M., Zhang, X.Y., Ren, S.Q. and Sun, J. (2015) Deep Residual Learning for Image Recognition. arXiv: 1512.03385.

[17] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.R. (2012) Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. arXiv: 1207.0580.

[18] Springenberg, J.T., Dosovitskiy, A., Brox, T. and Riedmiller, M. (2015) Striving for Simplicity: The All Convolutional Net. arXiv: 1412.6806.

[19] Prince, S.J.D. (2023) Understanding Deep Learning. MIT Press, Cambridge.

[20] Kingma, D.P. and Ba, J. (2017) Adam: A Method for Stochastic Optimization. arXiv: 1412.6980.

[21] Wei, W. (2020) Small Object Detection Based on Deep Learning. 2020 *IEEE International Conference on Power*, *Intelligent Computing and Systems* (*ICPICS*), Shenyang, 28-30 July 2020, 938-943. https://doi.org/10.1109/ICPICS50287.2020.9202185

Appendix. Architecture of Classical CNN Model

A1. Eighteen Convolution Layers with 2D Convolution and ReLU Activation

- ✓ Six max pooling layers at every third Convolution layer, with a stride of 2, downsizing the feature maps by half.
- ✓ With the preprocessed thermal image size of 512 × 640 pixels as input to the first Convolution layer, results in a feature map of size (batch = 32, channels = 8, height = 2, width = 2) at the last convolution layer.
- ✓ A fully connected layer with thirty-two inputs and two outputs.
- ✓ A softmax layer with two inputs and two outputs.

A2. Performance of Classical CNN Model without Regularization

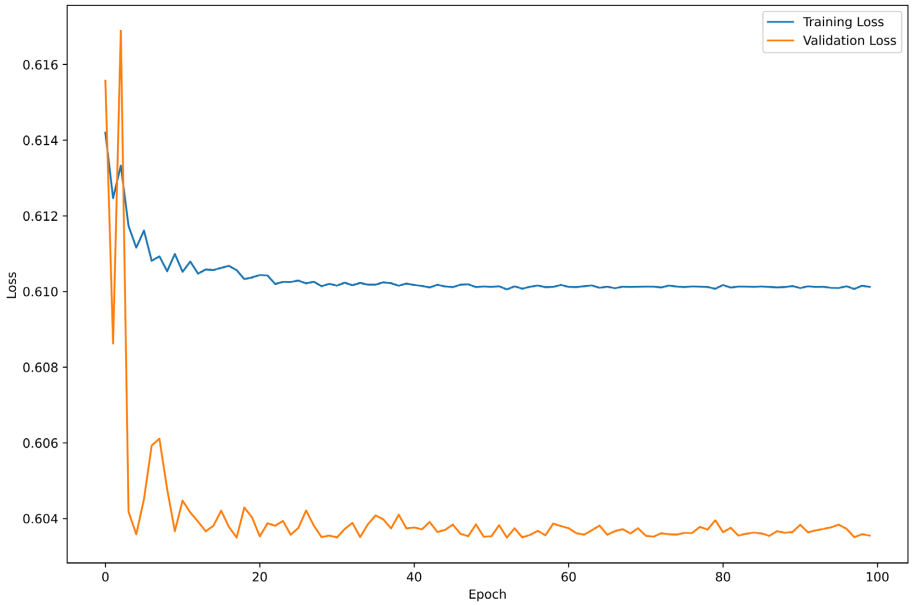


Figure A1. Training and validation loss of the classical CNN model without regularization.

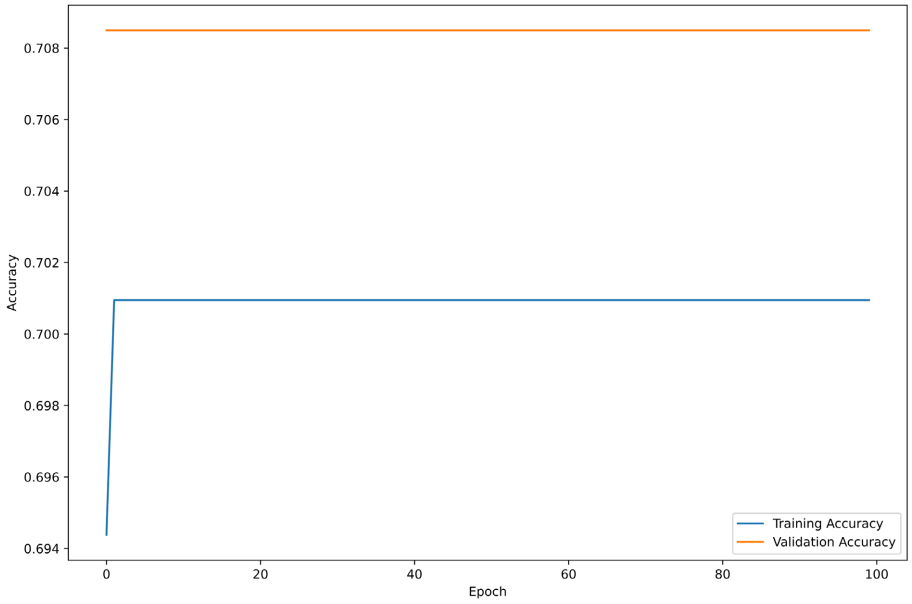


Figure A2. Training and validation accuracy of the classical CNN model without regularization.

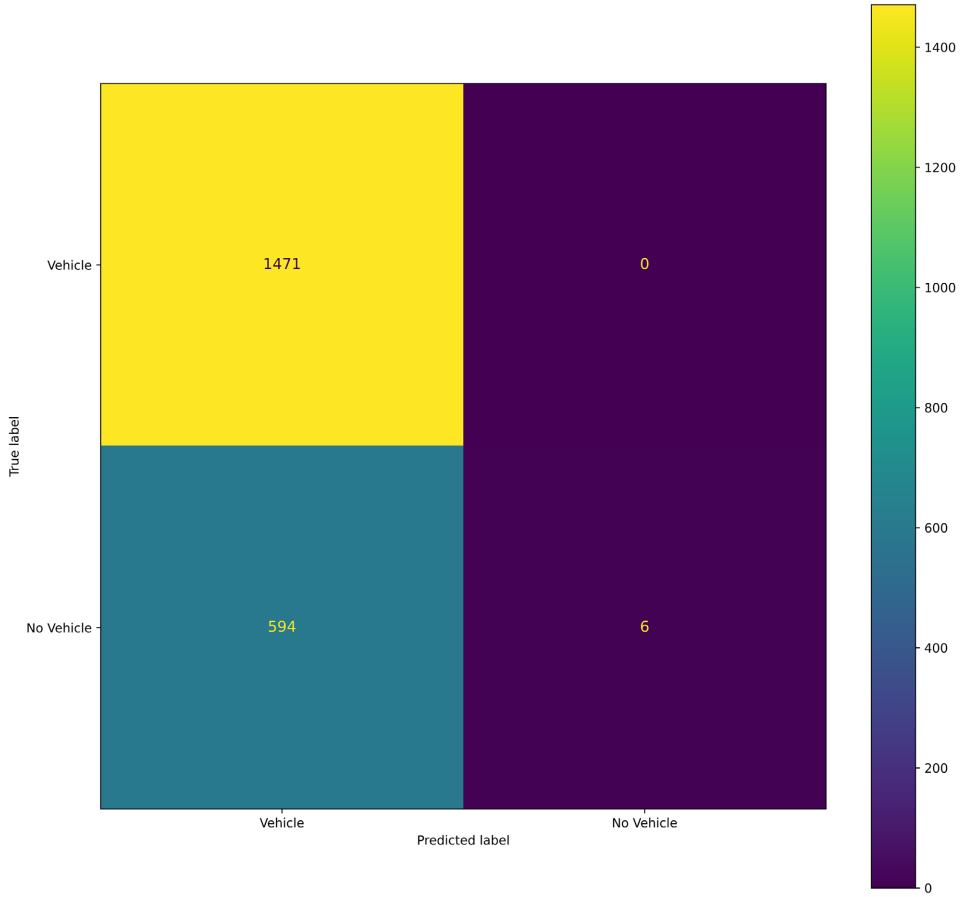


Figure A3. Confusion matrix of the classical CNN Model without regularization.

A3. Performance of Classical CNN Model with Regularization

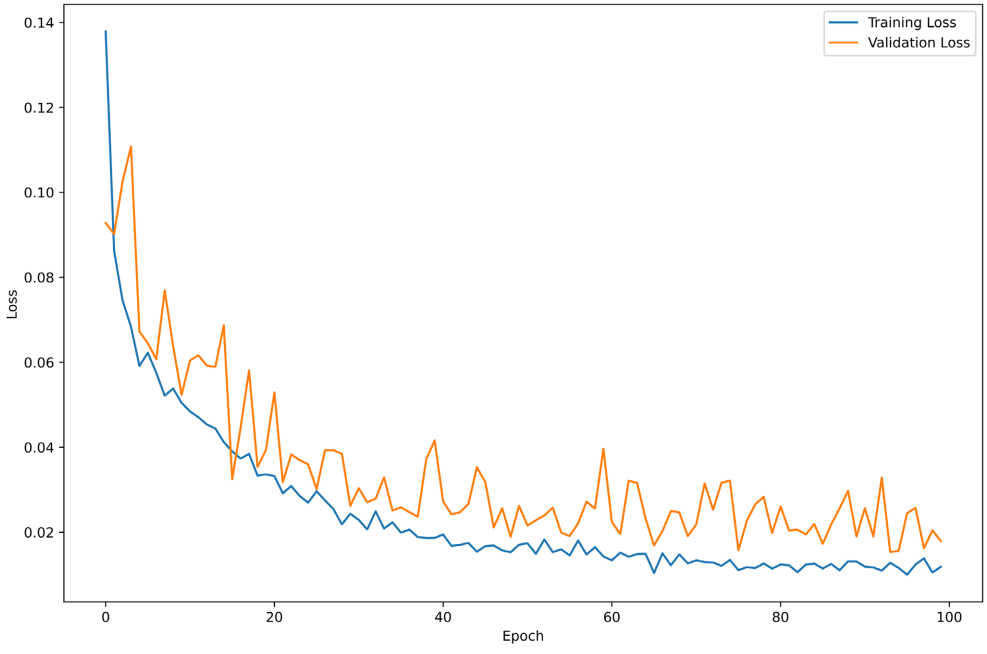


Figure A4. Training and validation loss of the classical CNN model with regularization.

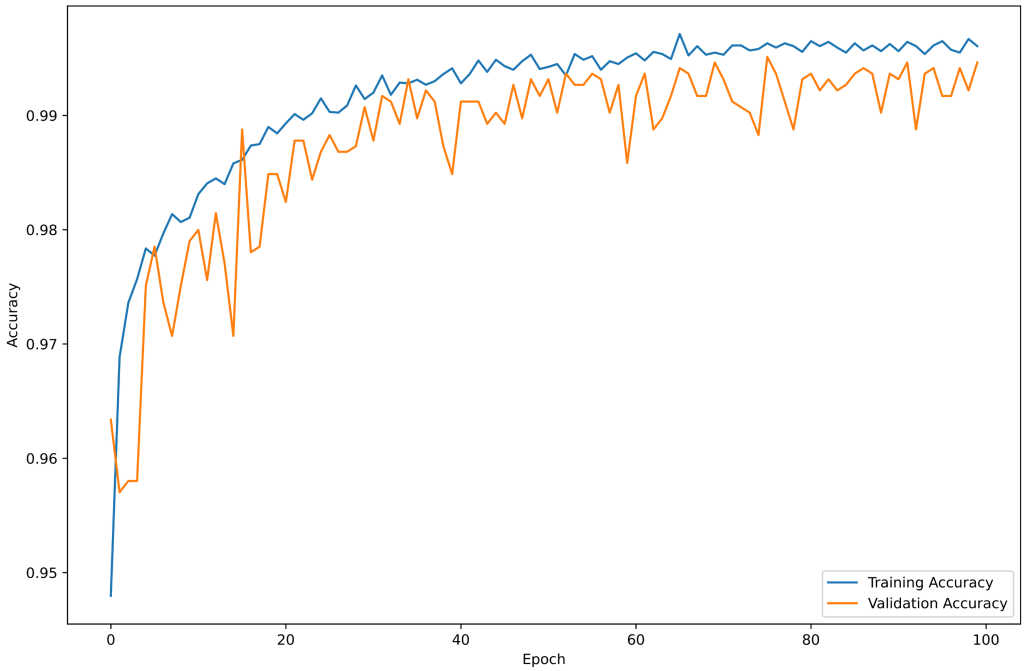


Figure A5. Training and validation accuracy of the classical CNN model with regularization.

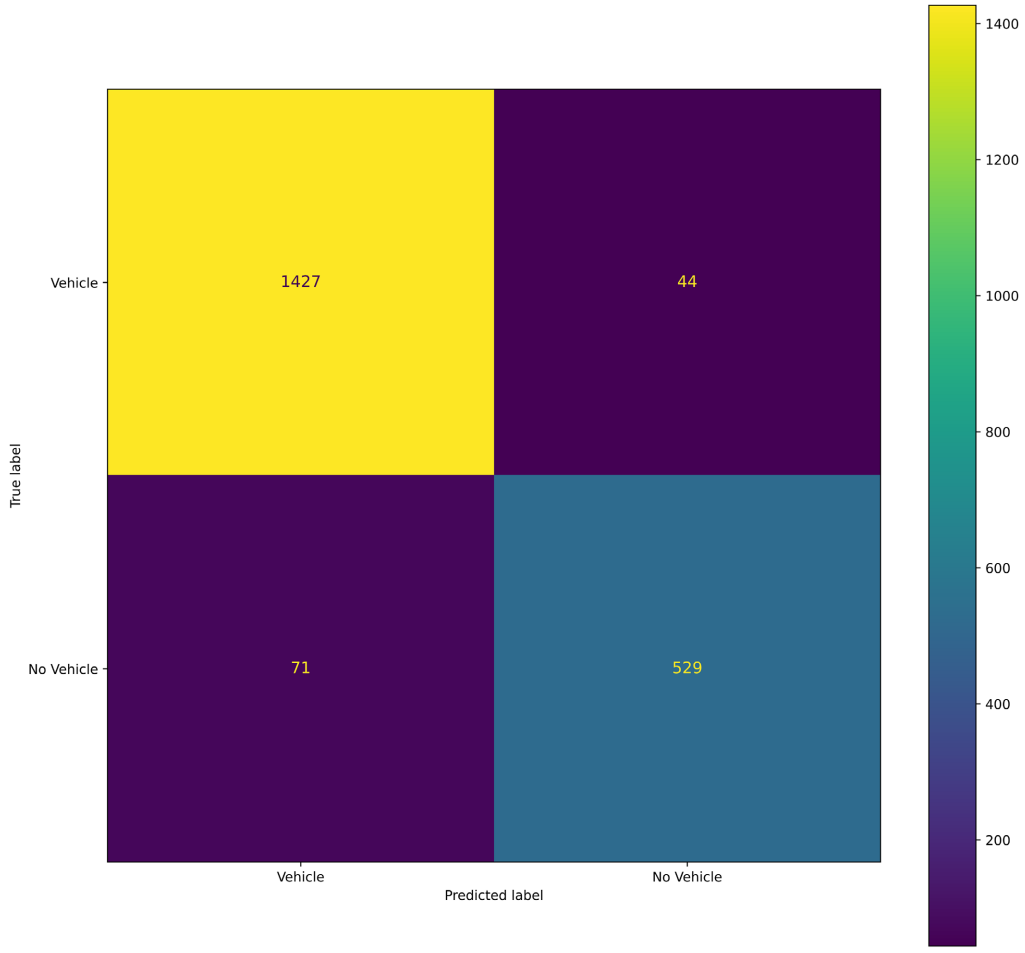


Figure A6. Confusion matrix of the classical CNN model with regularization.