UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

VISIBILITY ESTIMATION FROM CAMERA IMAGES USING DEEP LEARNING

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

MELISSA WILSON REYES
Norman, Oklahoma
2023

VISIBILITY ESTIMATION FROM CAMERA IMAGES USING DEEP LEARNING

A THESIS APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY THE COMMITTEE CONSISTING OF

Dr. Andrew H. Fagg (Chair)

Dr. Amy McGovern

Dr. Kara Sulia

Dr. Dimitris Diochnos

Acknowledgments

I would like to express my deepest appreciation to my committee, Dr. Amy McGovern, Dr. Kara Sulia, and Dr. Dimitris Diochnos, for providing me with encouragement and patience throughout the duration of this project. I would also like to extend my deepest gratitude to my advisor and chair of my committee, Dr. Andrew H. Fagg, who, over the past six years, has provided me with unparalleled mentorship and invaluable opportunities that have changed the course of my professional and academic career forever. His impact on my life cannot be underestimated. I cannot begin to express my thanks to Dr. Monique Shotande, who is an incredible role model to many including me. Over the past six years, she has provided me with unwavering and invaluable support, guidance, and friendship. I am also extremely grateful to Jay Rothenberger whose profound belief in my abilities and enduring encouragement has kept me going, even in the most difficult times.

I also had great pleasure of working with Carly Sutter, Arnoldas Kurbanovas, Aaron Evans, Dr. Vanessa Przybylo, Dr. Lauriana Gaudet, Dr. Nicholas Bassill, Dr. Christopher Wirz, Dr. Jerry Brotzge, and Dr. Christopher Thorncroft, all of whom provided invaluable guidance and expertise that made this work possible. I am also grateful to Vishnu Kadiyala, Dr. Randy Chase, Charlotte Cabrera, Max Sasser, and Chris Cepin for their time and collaboration on this project. Thanks should also go to the xCITE and OSCER Supercomputing folks. This work would not been possible without without their assistance and generous resources.

Also special thanks to Dad, Johnathon, Chachi, and the rest of my loving family, all of whom have always supported, and nurtured me and never failed to be an ear I could rely on.

This research is made possible by the New York State (NYS) Mesonet. Original funding for the NYS Mesonet was provided by Federal Emergency Management Agency grant FEMA-4085-DR-NY, with the continued support of the NYS Division

of Homeland Security & Emergency Services; the State of New York; the Research Foundation for the State University of New York (SUNY); the University at Albany; the Atmospheric Sciences Research Center (ASRC) at the University at Albany; and the Department of Atmospheric and Environmental Sciences (DAES) at the University at Albany.

This work is supported by the National Science Foundation under Grant No. ICER-2019758.

Table of Contents

List Of Tables

List Of Figures

Abstract

Atmospheric visibility is an important and complex meteorological variable that directly affects safe and reliable transportation. Specifically, declining visibility can pose an increased risk to automotive, aviation, and maritime traffic and operations. Traditional visibility sensors, e.g., those of the Automated Surface Observing Systems (ASOS) network, are costly and designed for air traffic use, thus these visibility sensor networks have limited coverage state-wide. In contrast, camera footage is highly available, accessible, and fairly inexpensive. While it is possible to construct a model that detects a visibility measure for a single camera or location, this type of model is not generalizable to new locations with varying physical features or different fields of view.

I propose a comparative visibility model that is generalizable solution to new locations. I train a convolutional neural network (CNN) that compares a query image and a reference image that originate from the same camera, and determines the degree to which the query image is less visible than the reference image. A query image from a new camera can then be compared to a set of reference images with known visibility distances from the same camera. These comparisons can then be used to infer the query image's underlying visibility distance. In addition, a model can be trained using a set of locations that have different maximum visibility distances, fields of view, and physical characteristics. The resulting comparative model can generalize to novel sites. When combined with a small number of calibrated reference images for a given site, visibility distances can be accurately estimated from previously unseen query images. Results from a large combined NYSM/ASOS data set show that the models learned using the proposed method are able to generalize to new locations. The approach is successful in the comparative case and the numerical visibility prediction case. With these outcomes, the model is also able to effectively monitor visibility over time.

Chapter 1

Introduction

Atmospheric visibility is a vital meteorological variable that greatly influences and impacts aviation, maritime, and automotive traffic. A reduction in visibility due to fog, air pollution, or severe weather can lead to accidents and travel delays. Visibility sensors, such as scatterometers or transmissometers, are powerful weather instruments that can detect and monitor visibility for a specific location. They are mainly deployed at airports to monitor visibility conditions for airplane take-off and landing. However, they are expensive and current weather networks that leverage visibility sensors have extremely limited coverage. For automotive and maritime applications, it is important that a visibility sensing network is deployed over a large geographical region and contains many sensors, e.g., statewide monitoring of highway corridors. Low cost video cameras and closed circuit television (CCTV) systems are becoming ubiquitous and available. Therefore, this work aims to detect and monitor atmospheric visibility from camera images alone.

Visibility detection from monocular camera images is a challenging task. Although humans make use of various monocular cues (e.g., semantic information) to estimate the depth or distances in a scene, there is no inherent depth information in a 2-D camera image. So, it can be difficult to identify the distance between different landmarks and the camera. Also, reduced visibility conditions with the same measurement can look radically different in the camera lens. Time of day, time of year, weather condition, camera position, and temporary objects, like cars, can affect how a scene looks in

reduced visibility conditions. Furthermore, there is a lack of publicly available image-based visibility data sets, since these data sets are propriety property or available cameras are not co-located with visibility sensors.

Previous work trained deep learning models to estimate visibility by classifying image visibility in "bins" or "classes" (Palvanov and Cho, 2019) or predicting a numerical visibility value (Zou et al., 2021). These models have not been explicitly shown to generalize to new locations. In addition, other work in the literature used deep learning models to predict a relative visibility estimate by comparing pairs of images (You et al., 2019; Xun et al., 2022). However, like other work, the efficacy of this approach has not been demonstrated with camera images from novel scenes or locations.

Following You et al. (2019) and Zou et al. (2021), I propose an approach with three major contributions. One, I use deep learning to train a model that predicts the comparative visibility between two images while ensuring that the models learned can generalize to novel sites and learn in the presence of significant noise in the labels. Two, I train these models using an orthogonalized cross-validation procedure, in which each rotation is temporally and spatially independent. Training, validation, and testing data are mutually exclusive sets of camera locations and years. Three, I translate a set of comparisons between a curated reference set and a novel query image to a numerical visibility estimate. The test set results show that the learned models can generalize to independent test data, even when there is a significant number of mislabeled examples in the training data.

There is a lack of robust visibility image data sets due to the high cost of visibility sensors and the difficulty of maintaining traditional sensor networks. In this work, I leverage two wide-spread and highly maintained weather networks, the New York State Mesonet (NYSM) and the Automated Surface Observing System (ASOS) in order to

build a comprehensive data set that includes multiple locations, multiple years, and varying weather phenomena.

Chapter 2 discusses how visibility is currently monitored, existing data sets, as well as previous approaches related to determining visibility from camera images. Chapter 3 provides details about existing sources of camera images and visibility measurements that were curated for this work. Chapter 4 outlines my novel approach to learn a comparative visibility estimator, as well as a method to translate comparison probabilities to a continuous or numerical visibility prediction. Next, chapter 4 documents how reference images are selected for each location. Chapter 5 details the results of the proposed comparative visibility estimator and the corresponding numerical visibility predictions. Finally, Chapter 6 highlights and discusses where the model performs well, the limits of the work, and how characteristics of the data set influence how the model learns and generalizes.

Chapter 2

Related Work

Automatic visibility estimation is a challenging problem that is studied for its various applications in aviation, maritime, and automotive traffic (Carley et al., 2021; Amani et al., 2020; Mathew and Pulugurtha, 2022). There exist many weather networks and monitoring systems to detect and report low visibility conditions, e.g., the Automated Surface Observating Systems (ASOS) program (National Oceanic and Atmospheric Administration, 1998) and NOAA's Physical Oceanographic Real-Time Systems (PORTS) (Edwing, 2019). However, these networks are limited in their coverage and deployment. Estimating visibility from images has become of great interest due to the broad coverage and accessibility of cameras, such as webcams or CCTV systems. Nevertheless, it is difficult to model visibility conditions from camera images in a generalizable way. Similar atmospheric visibility conditions can look completely different due to location, time of day, time of year, and the cause of low visibility conditions. Machine learning has shown to be a successful approach for modeling atmospheric visibility and can provide a unified approach for translating camera images to useful visibility predictions. In this chapter, I discuss the various causes of low visibility conditions, existing data sets that capture atmospheric visibility changes, and the use of computer vision and machine learning techniques to predict atmospheric visibility from camera images.

2.1 Visibility Monitoring

Atmospheric visibility is defined by the American Meteorological Society as "the greatest distance in a given direction at which it is just possible to see and identify with the unaided eye 1) in the daytime, a prominent dark object against the sky at the horizon, and 2) at night, a known, preferably unfocused, moderately intense light source" or "the clarity with which an object can be seen" (American Meteorological Society, 2012). Low visibility conditions can be caused by a wide variety of meteorological and physical phenomena. Common causes include fog, precipitation, snow fall, or other severe weather events. Because these weather conditions occur most often during the winter, low visibility conditions caused by weather is closely correlated to specific times of year. However, due to wide-spread environmental changes, atmospheric pollutants, such as smog or haze, can cause low visibility conditions year round.

Low visibility conditions, whether from naturally occurring weather phenomena or man-made pollution, pose a risk to all types of automotive, aviation, or maritime travel. Many U.S. states have safety guidelines for driving in foggy conditions (New York State Department of Motor Vehicles), and Mathew and Pulugurtha (2022) demonstrated that rainfall and low visibility had a negative effect on travel time reliability for roadway drivers. Low horizontal visibility can pose a risk to aviators, and is associated with aviation accidents (Carley et al., 2021). In addition, dense fog can be hazardous to maritime travel, and can disorient even experienced boaters (National Weather Service, b). Because of this, various agencies have developed standards for collecting and reporting visibility conditions.

2.1.1 Automated Surface Observing Systems

The Automated Surface Observing Systems (ASOS) program is a collaborative effort between the National Weather Service (NWS), the Federal Aviation Administration (FAA), and the Department of Defense (DOD) to provide support for meteorological research, weather forecasting, and aviation activities. The ASOS program is comprised of numerous weather stations distributed across the United States. Since the ASOS program has an aviation focus, these weather stations tend to be co-located with airports. These stations collect and monitor a wide variety of meteorological variables and phenomena, for example, precipitation, wind speed, and visibility (National Weather Service, a).

Visibility measurements are collected by a forward scatter sensor and reported in statute miles. Samples are collected at one-minute intervals, and a running harmonic mean is computed across the last ten minutes. This has a conservative effect when reporting visibility changes, which is preferable for aviation applications. When atmospheric visibility is dropping rapidly, the harmonic mean is more responsive and more quickly reports low visibility estimates. However, when atmospheric visibility is rising rapidly, the harmonic mean is slower to reflect high visibility, thus more conservative when reporting high visibility estimates (National Oceanic and Atmospheric Administration, 1998). Although the ASOS visibility measurements do not always represent the exact current visibility, they have been shown to be a reliable and reasonable reporting of atmospheric visibility.

The ASOS network provides quality detection of visibility because atmospheric visibility has a huge effect on how aviators operate. The FAA provides different flight categories and rules for how aviators operate based on the visibility conditions. Different flight categories are determined by the visibility in miles (Table 2.1). Aviators

Flight Category	Visibility (mi)
Visual flight rules (VFR)	$5 < x$
Marginal visual flight rules (MVFR)	$3 < x \leq 5$
Instrument flight rules (IFR)	$1 < x \leq 3$
Low instrument flight rules (LIFR)	$x \leq 1$

Table 2.1: FAA flight categories for visibility. (Carley et al., 2021; Bouhsine et al., 2022)

Marine Forecast	Nautical Miles
Good	$5 < x$
Moderate	$2 \leq x \leq 5$
Poor	$0.5 \leq x < 2$
Very Poor	$x < 0.5$

Table 2.2: International standards for describing reduced visibility in marine forecasts. (National Weather Service, b)

are able to use visual references during flight in higher visibility conditions, and are limited to aircraft instruments in low visibility conditions. Maritime travel is affected in a similar way and requires boaters to augment their operations based on the fog conditions. Although alike, there are some differences in visibility grading between aviation and maritime standards (Table 2.2).

2.1.2 The Mesonet

Although the ASOS weather stations provide invaluable sources of visibility estimates with nation-wide coverage, stations are usually limited to airports. In addition, there are no camera images associated with the ASOS network. However, in the United States there exist robust state-wide weather networks, also called Mesonets. The Mesonets are weather networks that were designed to facilitate mesoscale weather detection and mitigate the harmful effects of high impact weather events (Mesonet; NYS

Mesonet). Mesonet weather stations are equipped with numerous meteorological sensors that measure conditions such as temperature, wind speed, and relative humidity. These measurements are continuously collected at five minute intervals and monitored for quality control (McPherson et al., 2007; Brotzge et al., 2020).

The New York State Mesonet (NYS Mesonet or NYSM) was commissioned by the state of New York and the Department of Homeland Security and Emergency Services to tackle the devastating effects of hurricanes, blizzards, ice storms, tornadoes, and other severe weather. The network was designed, installed, and is currently operated by the State University of New York (SUNY) at Albany. The standard site network consists of 126 weather surface stations across the state of New York, with at least one station being placed in each county (Figure 2.1).

In addition to the various meteorological sensors deployed at Mesonet stations, each NYS Mesonet station is equipped with a small camera that collects images of the site. The station's camera is mounted on the weather tower approximately 2.5 meters above ground level and generally pointed northward. The angle of the camera is designed to capture the NYSM station gate entrance, the snow measuring stick, and ideally incorporate equal parts ground and sky in the image (Figure 2.2). The collected images are primarily used for data quality control, however, they are also used for monitoring other pertinent changes such as visibility, cloud cover, precipitation, vegetation cover, atmospheric phenomena, and even passing wildlife. However, many of these changes are monitored manually by individuals observing the camera feed from the different sites.

The images collected by the camera are captured and stored in RGB format during the "daytime" and grayscale format during the "nighttime." Daytime and nighttime are determined automatically by the camera using the magnitude of the light falling on

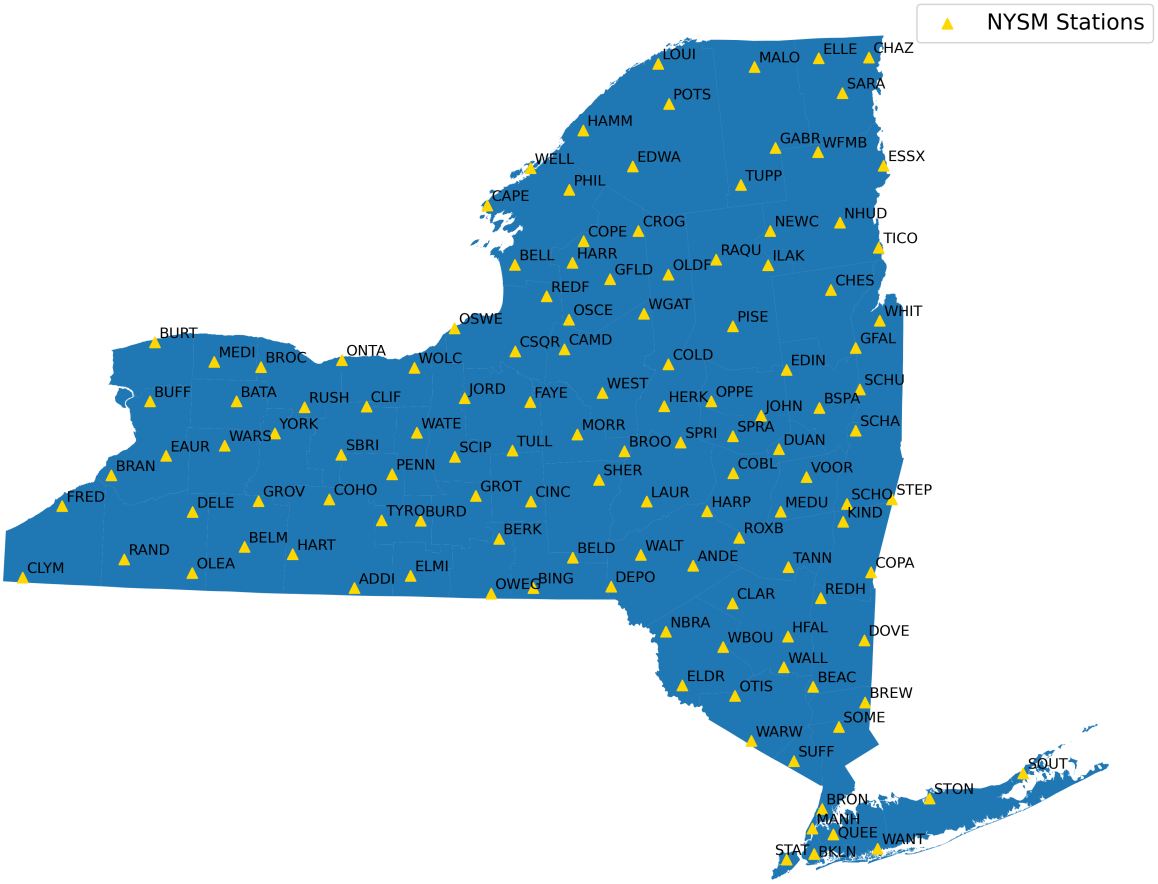


Figure 2.1: Map of New York state with NYSM standard site locations.

the camera's sensor. Moreover, the camera collects a RGB image when there is enough light in the lens and a grayscale image is taken using the camera's infrared capability when there is minimal or no light. RGB images are taken at five minute intervals during the daytime and grayscale images are taken at one hour intervals during the nighttime.

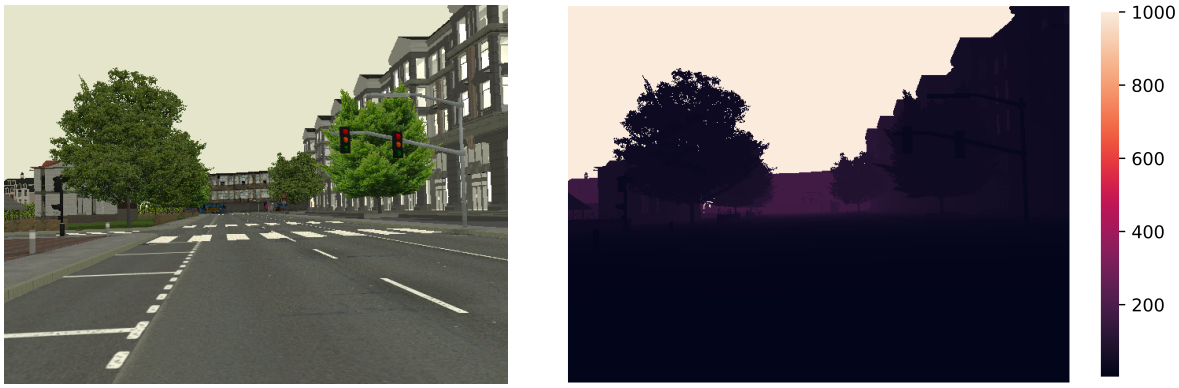
Figure 2.2: Glen Falls station in the October of 2020. The purple box highlights the gate entrance to the station and the green line represents the approximated horizon line.

2.2 Synthetic Visibility Simulations

Since fog has a hazardous effect on roadway conditions, visibility estimation is particularly important for camera-based driver assistance systems or autonomous self-driving efforts. A few synthetic data sets have been developed to support related efforts such as visibility enhancement, road segmentation, and depth estimation. Although originally designed to serve automated driving efforts, these data sets are used as standard bench-marking tools to determining the efficacy of image-based visibility estimation approaches.

The Foggy Road Images DAtabase (FRIDA) is a collection of computer simulated road environments, where artificial fog was applied to emulate different fog conditions (Tarel et al., 2012a). The formulation of Koschmieder's law (Middleton, 1952) used to generate the FRIDA data set is as follows:

$$L(u,v) = L_0(u,v)e^{-k\,d(u,v)} + L_s\left(1 - e^{-k\,d(u,v)}\right), \tag{2.1}$$



(a) (b)

Figure 2.3: Scene 3 in the FRIDA data set: (a) original scene and (b) the depth map in meters (Tarel et al., 2012a).

where $L(u, v)$ is the apparent luminance of the object, $L_0(u, v)$ is the intrinsic luminance of an object, $d(u, v)$ is the distance of the object at pixel (u, v), L_s is the luminance of the sky, and k is the extinction coefficient of the present fog.

In total, 18 unique scenes were captured in 640 x 480 color images and depth maps (Figure 2.3). Four different fog types were applied to each of the scenes to simulate a visibility distance of approximately 85.6 meters. "Uniform fog" (Figure 2.4a) was generated using Koschmieder's law to produce a visibility distance of exactly 85.6 meters. "Heterogeneous k fog" (Figure 2.4b) was similar except the extinction coefficient, k, was weighted differently depending on pixel position using Perlin's noise. Perlin's noise is a type of gradient noise often used in computer graphics to create image textures. The authors use Perlin's noise here to emulate the non-uniformity of real world fog. "Heterogeneous L_s fog" (Figure 2.4c) was where the underlying sky luminance, L_s, was weighted differently using Perlin's noise, as well. "Heterogeneous k and L_s fog" (Figure 2.4d) had both k and L_s weighted using two independent Perlin's noises. In total, the FRIDA data set is composed of 90 images, 18 original unaltered images and 72 images augmented with fog.

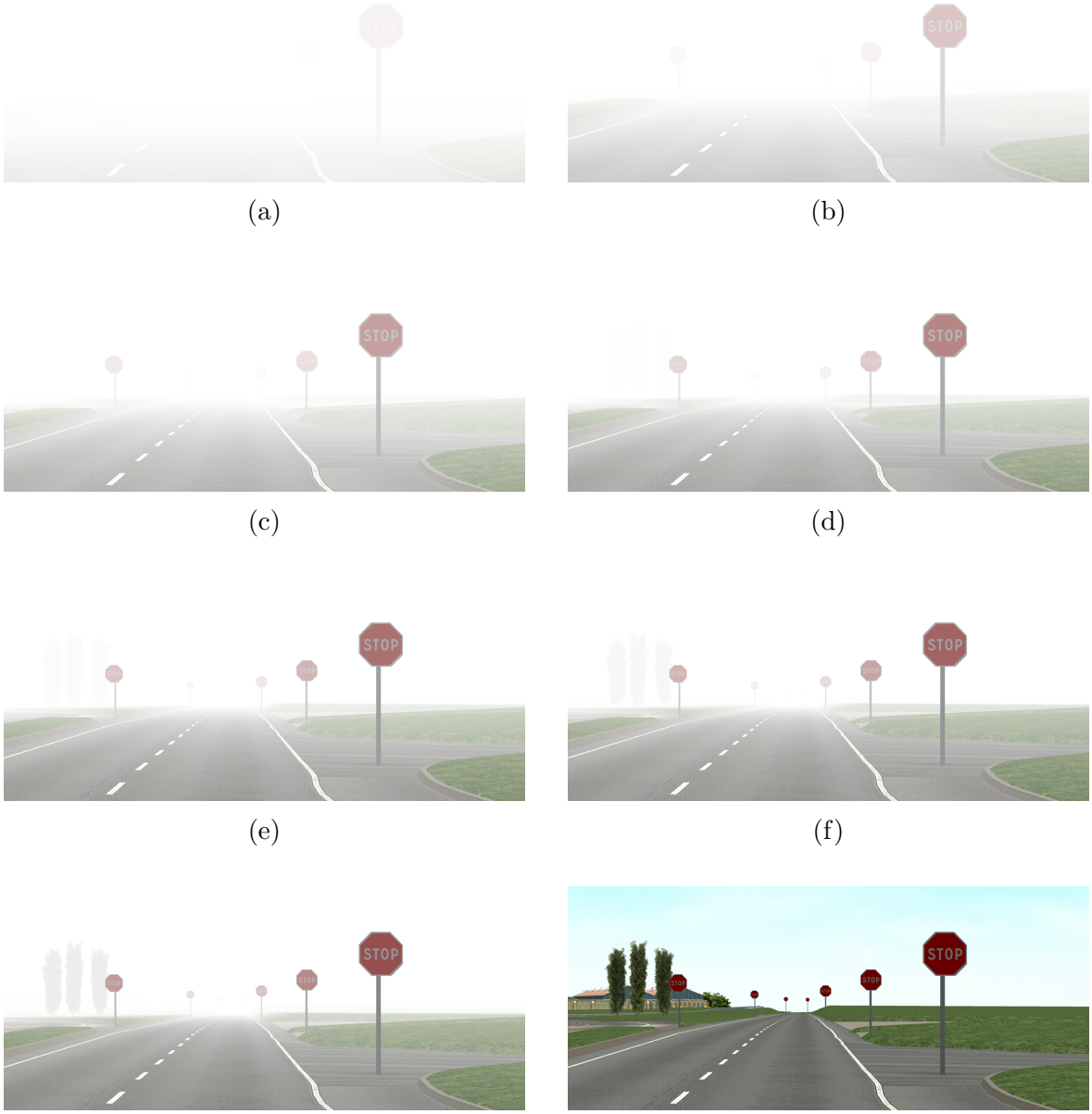
(a) (b)

(c) (d)

Figure 2.4: Different fog types applied to Scene 3 in the FRIDA data set, (a) Uniform fog, (b) Heterogeneous k fog, (c) Heterogeneous L_s fog, and (d) Heterogeneous k and L_s fog (Tarel et al., 2012a).

The second iteration of the data set, known as FRIDA2, consists of 66 unique roadway scenes (Tarel et al., 2012b). Like the original FRIDA data set, depth maps of the scene were included, as well as images augmented using the same four different types of fog used in the original FRIDA data set. However, for this data set the visibility is reduced to approximately 80 meters. In total, the FRIDA2 data set is composed of 330 images, 66 original unaltered images and 264 images augmented with fog.

The Foggy ROad Signs Images (FROSI) data set is another collection of synthetic computer generated scenes for visibility algorithms and research purposes. The FROSI data set is especially targeted toward self-driving or automated motor systems. As a result, the FROSI data set focuses on one distinct street-view scene. The variation in the data set comes from four sources: the different street signs, the camera angle and height, the camera position along the street, and the different levels of fog applied to the scene. Only uniform fog was applied to each of the images, however, the fog was applied in 8 gradations from 50 meters to full visibility (Figure 2.5). In the FROSI data set, there are 168 unique sign heights, and camera position variations. In addition, there are three different road signs and 8 visibility gradations. In total there are 4032 distinct images in the data set.



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

Figure 2.5: Different fog levels applied to a static scene in the FROSI data set, (a) 50 meters, (b) 100 meters, (c) 150 meters, (d) 200 meters, (e) 250 meters, (f) 300 meters, (g) 400 meters, and (h) no fog (Belaroussi and Gruyer, 2014).

2.3 Visibility Estimation from Camera Images

While there exist many efforts to monitor and record visibility using sensors or human observations, these methods are restricted by the locations of the sensors or require man-power for manual labeling. These limitations negatively impact the availability of large-scale coverage of visibility estimates. In order to increase broad access to visibility estimates, there has been a strong and consistent research effort to estimate visibility from camera images. The estimation can be framed as a regression problem, where models predict the visibility distance depicted in an image, or as a ordinal classification problem where models determine the visibility range that the image falls into.

2.3.1 Traditional Computer Vision Techniques

Traditional computer vision techniques mathematically represent features in order to translate inputs to outputs in a direct and efficient algorithmic way. Low visibility conditions can cause significant visual changes to the appearance of a scene. This is especially present in camera images, where visual landmarks can be altered or completely obscured by fog or pollution. Hallowell et al. (2007) describe an edge detection approach wherein a Sobel Filter is used to extract the edges of buildings, structures, and landmarks of scenes. A Sobel Filter is often utilized in computer vision for edge detection and image segmentation algorithms. It emphasizes or strengthens regions of high spatial frequency in the image, which correspond to edges. Their approach was evaluated on camera images from the Utah Department of Transportation (UDOT) with corresponding ASOS visibility measurements.

A composite image is first created to represent a highly visible clear day scene. This composite image is the running average of the edges extracted from the most recent clear day images. The composite image is used to remove "unexpected" edges

from incoming images. This process removes edges associated with traffic, snow piles, leaves, etc. From there, the magnitude of the edges in the image is calculated using a Fast-Fourier Transform and then summing the absolute value of all the pixels in the image. This magnitude has a linear relationship with visibility, so if this magnitude is high, the visibility is high, if the magnitude is low, the visibility is low. The authors use a linear model in order to translate input image's magnitude into a visibility distance estimate. Visibility is binned into 4 different classes (<1 mile, 1-5 miles, 5-10 miles, and $\geq$ 10 miles). The critical success index (CSI) is computed for each of the ranges, as defined below:

$$CSI = \frac{TP}{TP + FP + FN} \ , \tag{2.2}$$

where true positive, TP is the number of images correctly classified as in the target visibility bin, false positive, FP is the number of images that were falsely identified to be in the target visibility bin, and false negative, FN is the number if images that were falsely identified to be outside the target visibility bin. The CSI over all the visibility bins was 61%, however, images with a visibility of less than 1 mile had a CSI of 53%. One of the main reasons for their poor performance on the Utah DOT data set was the frequent shifting of camera position which would cause misalignment between the composite image's edges and input image's edges.

Carley et al. (2021) produced a similar algorithm, called Visibility Estimation through Image Analytics (VEIA), in order to determine the visibility distance estimate and FAA visibility class from a camera image. Like Hallowell et al. (2007), the VEIA algorithm uses a Sobel Filter to extract edges. The composite image is again a composition of recent clear day images. The algorithm measures the ratio of the sum of extracted edges from the input image to the sum of extracted edges from the composite

image. This ratio passed into a linear model to produce a visibility prediction in miles for the input image. The VEIA algorithm was developed using images from the FAA network of camera sites in Alaska and METAR visibility observations. METAR, also known as METeorological Aerodrome Reports, is a format used for reporting weather conditions that is often used by aircraft pilots and meterologists. Visibility was binned into the four different FAA flight categories (See Table 2.1). The VEIA algorithm performed well on images depicting high visibility conditions, VFR or MVFR, with a CSI greater than 85% and 60% respectively. However, the algorithm struggled with images depicting low visibility conditions, LIFR, with a CSI of less than 40%.

Statistical image-processing techniques from de-hazing efforts can be used to identify and describe visibility phenomena in camera images. He et al. (2011) observed that most non-sky regions of clear-day outdoor images often had pixels with very low intensities in at least one color channel. The amount of these "dark" pixels could be used to characterize the amount of light in the scene that is not scattered, but instead that reaches the camera. This pixel intensity measurement, also known as the "Dark Channel Prior" (DCP), can be calculated horizontally over various sub-images of a scene in order to infer the transition between ground and sky in the image. DCP was originally designed for de-hazing images, however, it can also be used to identify and characterize the amount of fog in an image. In clear images, the DCP changes sharply, going from regions containing ground to regions containing sky due to the clear horizon line (Wauben and Roth, 2016). However, in foggy images, where the horizon line is unclear or obscured, the DCP changes slowly and smoothly from the ground to sky sub-regions.

Edge detection can be used in combination with DCP in order to calculate global image features for visibility classification. Zhao-zheng et al. (2009) used an edge detection like Hallowell et al. (2007) however they focused on specific visual targets, such as

road lane boundaries. The atmospheric contrast between objects and the background were used to fit a non-linear model in order to a estimate a visibility distance. Wauben and Roth (2016) explored different methods for estimating visibility from camera images. One approach described was landmark discrimination where landmarks with known distances from the camera are identified. Edges are extracted from novel input images and clear day images using a Gaussian convolution. A threshold for edge strengths is empirically determined from the clear day images. If edges from landmarks in the input images do not meet this threshold, those landmarks are determined to be non-visible. The visibility range for the overall scene is deduced from the maximum distance to any visible landmark.

However, this approach requires input images to be labeled with landmarks and their respective distances, so authors also explored extracting global image features, such as the average number of edges present in the image, the DCP ground-to-sky transition or change point, and average brightness. Two methods were trained and evaluated on a set of images from the same location. A decision tree was fitted using the average number of edges and DCP change point and determined the presence or absence of dense fog (fog less than 250 meters). The decision tree achieved a probability of detection of 100% and a false alarm rate of 6% on the evaluation set. A regression model was fitted with the addition of average brightness. When distinguishing between the presence or absence of dense fog, the regression model was able to correctly identify dense fog with a probability of detection of 98% and a false alarm rate of 18%.

Image processing techniques and traditional physics-based models can be used in combination to identify and describe visibility phenomena in camera images. Information entropy theory has been adapted for image-processing to measure fog and haze in expressway corridors (Cheng et al., 2018). Furthermore, the extinction coefficient in

Koschmieder's law can be empirically estimated from a distribution of images. The extinction coefficient can then be used in combination with the DCP in order to aid scene visibility estimation (Li et al., 2019a; Yao and Huang, 2022). However, the extinction coefficient can be sensitive to lighting conditions and surface reflection, therefore Babari et al. (2012) use the texture contrast between various objects in the image to account for the variations of illumination in the scene. However, it is not clear if these approaches can generalize well to novel locations with completely different compositions or fields of view.

Some approaches take a preemptive approach to improving camera-based visibility estimations. Qin and Qin (2020) used a laser to measure current atmosphere transmissivity in order to improve visibility predictions from camera images. Wang et al. (2020) introduced three boards to be within frame of camera images. These target boards exuded difference color luminance values, and were used to improve applications of Koschmieder's law by standardizing the inherit brightness and luminescence of objects in the scene. Although useful, these approaches require additional equipment and maintenance for deployment.

2.3.2 Non-Convolutional Deep Learning Approaches

Extracting simple image features is not enough to handle the complex variation of low visibility conditions. High spatial frequencies are not the only cue that distinguishes visibility conditions. The research community has leveraged advancements in deep learning and computer vision to automatically extract complex features and robustly detect visibility from camera images.

Chaabani et al. (2017) explored using an artificial neural network (ANN) in order to predict numerical visibility ranges from camera images. The authors compressed images into a feature vector image descriptor, using a Fourier Transform to capture the

power spectrum of the image. Subsequently they apply Principal Component Analysis (PCA) to produce a vector of 201 values as input to the model. They trained and evaluated their model on the FROSI (Belaroussi and Gruyer, 2014) synthetic roadway data set. They compared their approach to a formulaic technique of comparing minimum of maximum pixel intensities of the image. In the end, their approach achieved 90.2% accuracy, while the formulaic approach achieved 62% accuracy.

2.3.3 CNN-Based Approaches

Convolutional Neural Networks (CNNs) are a specialization of ANNs that have been widely successful for computer vision applications. They are a powerful set of architectures that have demonstrated an ability to extract useful patterns from image data. CNNs use filters with learned weights in order to extract pertinent spatial features in the input. Subsequent layers of convolution hierarchically combine the spatial features together, extracting fundamental curves, corners, and shapes. While simple image filters, such as the Sobel filter, are able to extract useful edges of a specific spatial frequency, the complex phenomena in RGB images require a more robust framework for extracting patterns within image data. CNNs provide the structure for learning sophisticated features from diverse and varied data sources. Due to the continued success of CNNs, they are also used for developing models that can determine visibility from camera images.

Cho and Palvanov (2019) trained a CNN to grade the visibility of daytime images from seaside surveillance cameras in South Korea with respect to 21 different visibility classes. These classes range from 0 to 20,000 meters with 1000 meter increments. The CNN tested for this approach was quite shallow with only two layers of convolution. The authors reported the shallow architecture was a design choice to avoid over-fitting to the data set. Additionally, various forms of image augmentation were applied to

the input images, such as image rotation, flip, translate, and zoom, as well as, others modifying contrast, brightness, sharpness, and color. Their model achieved over 86% accuracy on the training set. There was no reported accuracy on the test set.

Although CNNs are useful on their own, specific data augmentation techniques can provide additional benefit to the model. In Palvanov and Cho (2019), the authors observed that the edges of landmarks in an image can be extracted using a Fast Fourier Transform (FFT) filter, and the fog or haze within the scene can be characterized mostly in the blue color channel of the RGB images by using a Spectral filter. They pipe the original image, the FFT filtered image, and the 2-D Spectral filtered image as three different branches of input into their CNN. The output of their model has a number of categorical classes that represent visibility ranges instead of a numerical visibility estimate.

Two different data sets were used for training and evaluating their models. The synthetic FROSI data set was used as a benchmark, and an additional real-world data set, Foggy Outdoor Visibility Images (FOVI), was collected to demonstrate the real-world application of their approach. The FOVI data set was curated from more than 3 million CCTV camera images with associated visibility values from co-located visibility sensors. These images where collected over 20 years and originated from 26 observations stations of the South Korea Meteorological Administration. The FOVI data set was further refined into a short-range and long-range task characterizing distances up to 1,000 meters and 20,000 meters respectively. The training, validation, and test sets were randomly partitioned with 70%, 10%, and 20% of the data respectively. This approach achieved a 94% accuracy on the FROSI test set, a 91.3% accuracy on the long-range test set, and a 89.5% accuracy on the short-range test set. Like many other approaches, since the training, validation, and test partitions are randomly assigned, these accuracies represent how well the model can perform on locations that it has seen

before. Due to this, there is no indication whether this approach can generalize to new locations.

Zhang et al. (2023) highlight the relationship between the depth of scene, how the fog degrades the image, and the scene visibility itself. The authors integrate physical constraints into their model in order to assure a robust solution. The authors develop a three part model called DQVENet that consists of a Transmission Estimation Module, a Depth Estimation Module (DEM), and an Extinction Coeffient Estimation Module. The Transmission Estimation Module (TEM) estimates the transmission map of a single image. When modeling atmospheric scattering, the transmission map describes the portion of light that is not scattered by atmospheric particulate and reaches the camera (Cai et al., 2016). The transmission map is important to image de-hazing techniques, however, in this case, the TEM module models the impact of fog on the image. The Depth Estimation Module (DEM) extracts the depth of the image at every pixel and models how far away certain landmarks are in the scene. Both the TEM and DEM are built using a U-Net based approach. U-Net is a convolutional neural network that was originally designed for image segmentation, however, has been successfully applied to other tasks such as image reconstruction or pixel-wise classification (Ronneberger et al., 2015; Mizusawa et al., 2021; Pan et al., 2020). Once the transmission map and depth map for a scene are computed from the TEM and DEM modules, the maps are concatenated and passed through the Extinction Coefficient Estimation Module (E3M) to predict a final visibility value. E3M consists of a pre-trained DenseNet (Huang et al., 2016), with the last classification layer replaced with a regression layer.

For the training data, the authors used 3,236 manually cleaned traffic images from high-speed road cameras. These images were labeled with visibility observations from the nearest meteorological station. For each camera and meteorological station pair,

the sites were within approximately 500 meters of each other. The DQVENet is trained using three steps. TEM and DEM modules are pre-trained using an image de-hazing data set and a depth estimation data set, respectively. The TEM and DEM modules are then frozen and combined for training the E3M module. In the final step, the entire DQVENet is fine-tuned with a small learning rate. The authors reported the model's performance using the correlation between the ground truth visibility distances and the model predicted visibility distances. Overall, the model achieved a correlation of 0.7237.

Numerical visibility prediction is a challenging task, especially if input images are from previously unseen locations. It is difficult, or at times impossible, to extract the dimensions or distances of objects from a 2D image accurately. Instead of training a model to learn absolute visibility, You et al. (2019) trained a model to learn the relative visibility of two images. The authors proposed a CNN-RNN model where the CNN learns the overall visibility of a image and the RNN (or Recurrant Neural Network) extracts the area of the image that depicts the furthest discernible distance present in the scene. The CNN-RNN module outputs a feature vector, and these feature vectors are used by a support vector regression machine to output a relative visibility score between 0 and 1. The model was trained on over 30,000 online curated images with human annotations as to which images were more or less visible than the other. In the final evaluation, the model was fine-tuned on a small real-world data set of 3,146 images, where 60% of the images were used for training and 40% of the images were reserved for testing. The model achieved a test set regression error of 0.19 kilometers.

2.3.4 Transfer Learning Approaches

Reliable and robust visibility image data sets are difficult to create and acquire. Some researchers turn to transfer learning as a way to combat the limitations of small data

sets. Li et al. (2017) used expert visibility observations from the Hong Kong Observatory and images from a nearby web-cam to create a database of 1,003 visibility-labeled images; 703 of the images were selected for training and the rest were reserved for testing. In addition, images were cropped to highlight areas of the images that captured landmarks that were furthest away from the camera. The authors train a two-part model. Firstly, they used AlexNet pre-trained (Krizhevsky et al., 2012) on ImageNet as a feature extractor. Secondly, they used a generalized regression neural network (GRNN) to produce the final numerical visibility prediction. The authors reported an accuracy of 77.9% on the training set and 61.8% on the test set. They suggested that the poor performance of the model was due to the low image resolution after cropping.

Li et al. (2019b) also collected data from the Hong Kong Observatory. In total, the authors used 5,511 images and corresponding visibility measurements collected by a forward-scattering visibility meter. These image and visibility pairs were used to train and evaluate a model to produce visibility predictions. 4,078 images were randomly selected for training and 1,433 were reserved for testing. The authors use a pre-trained image quality assessment deep convolutional neural network (Bosse et al., 2018) as a feature extractor. Each image is split into 9 sub-regions and processed by the feature extractor. The feature encoding for each of the sub-regions are then fused with Support Vector Regression in order to predict a numerical visibility value for the image. In this work, the model was more successful with an overall correct detection rate of 93.86% on the test set.

Li et al. (2021) also used data from the Hong Kong Observatory and built a database of 4,841 images with corresponding visibility measurements from expert observations. Images were pre-processed by applying gray averaging, gaussian blur, and an adaptive threshold to segment each image. Finally, each image is split into 5 different sub-regions which are processed by a pre-trained feature extractor. The extracted features are then

to passed to a Support Vector Regression model to finally output a numerical visibility value. The authors tested a variety of pre-trained networks such as VGG-16 (Simonyan and Zisserman, 2014), VGG-19 (Simonyan and Zisserman, 2014), DenseNet (Huang et al., 2016), and ResNet50 (He et al., 2015). Like Li et al. (2017) and Li et al. (2019b), the training and test sets were randomly divided resulting in 3,630 images selected for training and 1,211 reserved for testing. In the end, the authors evaluated their results by partitioning the visibility distances into 5 bins (0-10km, 11-20km, 21-30km, 31-40km, and 41-50km). The authors report an overall accuracy of 90.32% on the test set.

In these approaches, transfer learning is demonstrated to be a useful tool in the absence of a larger and robust data set. Authors managed to achieve over 90% accuracy on less than 2 months of training data (Li et al., 2021). However, across these studies only one location was used per data set. In addition, these data sets were manually cleaned. Often times the training, validation, and test sets were randomized, so there was no temporal independence between the sets. This limits the applications of this work where results are not demonstrated on novel years and restrict the deployment of these models to specific locations.

In some domains, such as aviation, categorical visibility classes are more informative than numerical visibility values. In Bouhsine et al. (2022), various deep learning approaches were used to predict the FAA visibility conditions (Table 2.1). The authors fine-tuned three models pre-trained on ImageNet (Deng et al., 2009) (VGG16, ResNet50, and DenseNet) and a Vision Transformer built and trained from scratch. The data set was collected by the FAA and consists of 27,746 images. Like many visibility data sets, it suffered from a lack of low visibility examples (LIFR). The data was divided in 80/20 split for training and validation. DenseNet was the best performing model on the validation set with a 98.5% accuracy, however, the vision transformer

provided similar results. Like many of the data sets collected for visibility applications, there is autocorrelation between the training, validation, and test sets, so high accuracy is expected. Like the other transfer learning methods, high accuracy may suggest good performance on locations already represented in the data set, however, does not imply generlizability to new locations.

Mi et al. (2020) optimized YOLOv5 (Jocher et al., 2022) to automatically detect the visibility of a camera video. In their method, YOLOv5 is used to identify landmarks in the camera frame, then their method uses a camera imaging model to calculate the distance of these landmarks, and finally determines the visibility estimation from the distance of the furthest identifiable landmark. The camera imaging model requires prior information about the camera site, such as the height and angle of the installed cameras. The data set was comprised of 2,000 images from a single city airport and were divided into a training and testing set with 80% and 20% of the data respectively. The model was able to achieve a recall rate of 100% and a precision rate of 90% on the training set. Like other approaches, this architecture relies on prior knowledge of a given scene. Even more so this approach requires information about the camera deployment itself.

Many approaches show impressive results on their respective test sets. The models appear to learn how to distinguish between different visibility conditions. However, many approaches randomize their data sets which caused temporal autocorrelation between the training, validation, and test sets. This could be problematic when camera images are sampled frequently, such as every five minutes. Many approaches use images from cameras that were deployed with the intention of capturing and monitoring weather phenomena. Often times, the scenes and locations captured by these cameras do not change substantially in composition, so extremely similar examples could be represented in the training, validation, and test sets. In addition, low visibility

conditions are usually quite uncommon compared to the frequency of high visibility conditions. For example, in Li et al. (2019b), the test set had 1007 images that had a visibility distance between 2 and 10 kilometers, however, only had 39 images that had a visibility distance between 0-0.5 kilometers. In addition, low visibility examples could come from the same weather events, so the models may be over-fitting to specific low visibility events, rather than achieving a generalizable notion of what low visibility looks like. Furthermore, the vast majority of these approaches do not evaluate their approach on images from a completely novel locations. These approaches demonstrate good performance on previously seen locations, however, their generalizability to new locations remains untested.

Chapter 3

Data Set Collection & Analysis

When estimating visibility from camera images it is important to have a well-defined robust data set that encompasses multiple locations and various visibility conditions. In previous work, independent groups have accessed existing meteorological data sets or sourced and labeled new data sets. However, many of these data sets are privately available, focus on short-range visibility tasks, consist of only one location, or contain only a few thousand labeled images (Bouhsine et al., 2022; Belaroussi and Gruyer, 2014; Palvanov and Cho, 2019). Due to the lack of available visibility-labeled camera image data sets, I utilize a combination of publicly available and privately available data sources to build my own.

In New York State, ASOS towers monitor and record visibility measurements across the state, but these towers do not have cameras. Conversely, the New York State Mesonet stations collect camera images of its site, however, do not have dedicated visibility sensors. In this section, I will describe how I synthesize individual data sources to create a unified visibility-labeled camera image data set for deep-learning model building, training, and evaluation.

3.1 Automated Surface Observing System (ASOS) Visibility Measurements

As described in Section 2.1.1, the ASOS network is a vital and useful weather network targeted toward aviation applications. Due to the hazard of low visibility conditions, the ASOS network collects and provides quality visibility measurements at many of its sites. These visibility measurements are useful for performing automated labeling of camera images taken nearby. In this section, I will discuss how I access the ASOS visibility readings and how I pre-process them to be automated labels for camera images.

Sensor readings and measurements from the New York State ASOS network are publicly available online (Iowa State University). There is data from 52 different ASOS stations available for download. Some stations have visibility reports going as far back as 1949. Over the years some stations have been decommissioned, leaving 49 of them currently active, as shown in Figure 3.1. Since the first New York State Mesonet station was commissioned in 2016, I removed all ASOS stations that were decommissioned before 2017. This filtering step assures that all visibility data occurred in the same time period as the NYS Mesonet data.

Unlike the regular sampling of the NYS Mesonet, the frequency of visibility measurements can vary widely. The New York City (NYC) ASOS station has visibility measurements for every hour of the day and the Potsdam (PTD) ASOS station has visibility measurements for every 20 minutes. However, many stations have visibility measurements for every 5 minutes. The data is already cleaned and missing values are easily filtered out. However, it is is important to recall that the ASOS network does

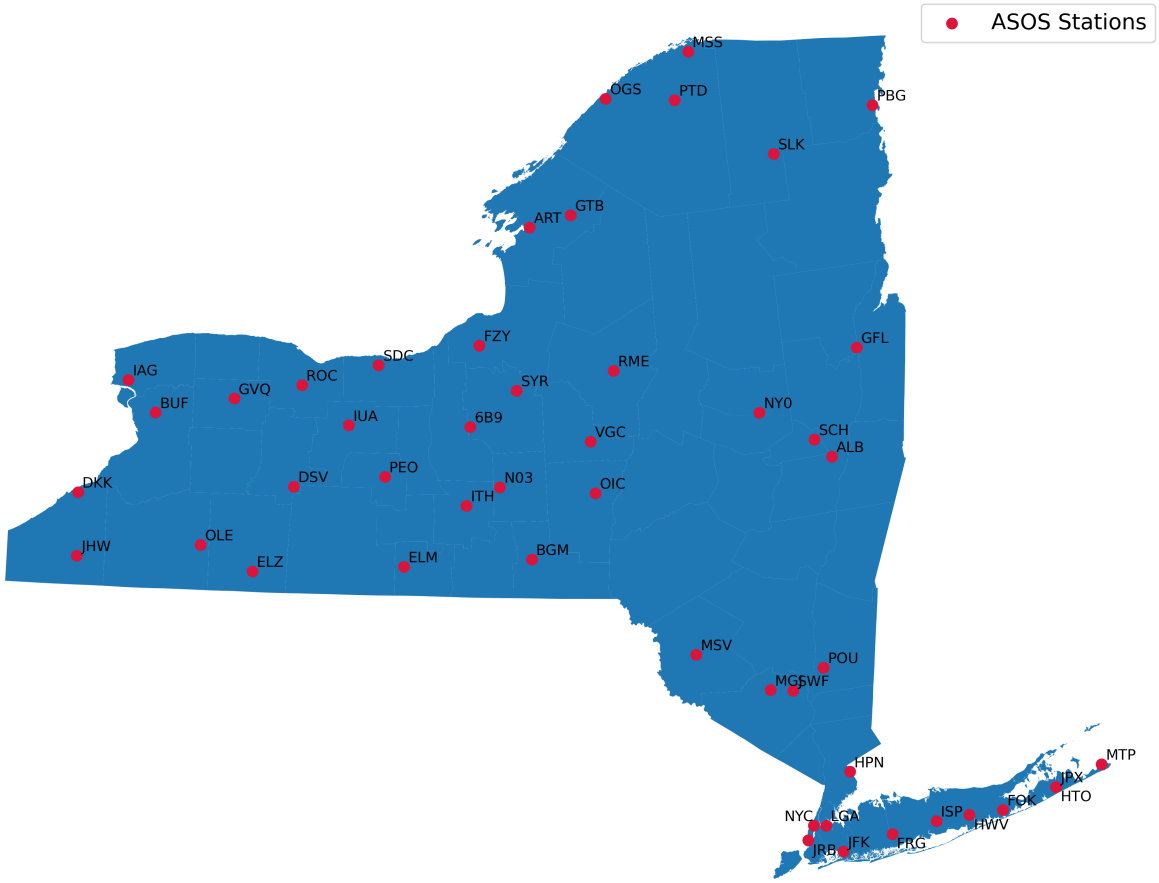


Figure 3.1: Active ASOS sites with publicly available data.

not equip cameras for each of their towers. This data set is used in conjunction with other image data sources like the New York State Mesonet.

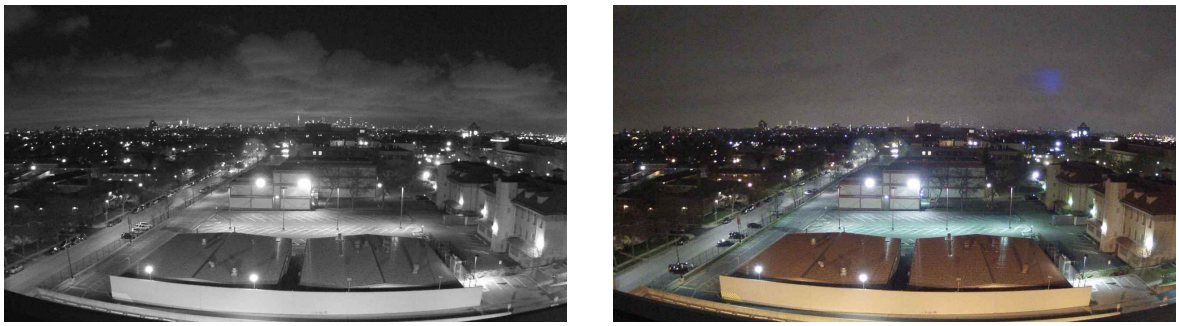
3.2 New York State Mesonet Camera Images

As described in Section 2.1.2, the New York State (NYS) Mesonet is a large and robust weather network that monitors state-wide weather conditions. Although the NYS Mesonet collects data on a wide array of meteorological phenomena, the NYS Mesonet towers are not equipped with a visibility sensor. However, the images that are collected are high-quality and provide the basis for this work's data set. In this section I will describe how I process the data set in preparation for model training and evaluation.

3.2.1 Ambient Light and Image Mode

As described in Section 2.1.2, RGB images are generally designated to be "daytime" and grayscale images are generally designated to be "nighttime". However, there are some stations whose data deviate from this rule. For these stations, RGB images do not always represent daytime images and grayscale images do not always represent nighttime images. Hyper-urban stations, such as the one located in Queens, have activity and lights in view all day and night. For the Queens station, in particular, there is rarely a time where the scene gets dark enough to trigger the infrared mode. As demonstrated in Figure 3.2, a true nighttime image for the Queens station can look very similar to an image with daytime mode conditions.

Conversely, if a camera produces a grayscale image, it does not guarantee that it is a truly nighttime image. As seen in Figure 3.3, the left image was taken approximately one hour after sunrise, however, it was collected as a grayscale image. The right image is from the same station on the same day, only collected five minutes later. In



(a) (b)

Figure 3.2: Queens station in the April of 2017. Both of these images were collected at 12:00am local time, however image (a) was collected two days before image (b).

(a) (b)

Figure 3.3: Elmira station in January of 2018. Image (a) was collected 8:35am local time and image (b) was collected five minutes later at 8:40am local time.

this situation the camera's sensor does not work exactly as expected and occasionally captures daytime images in grayscale.

Although the NYS Mesonet stations collect images during the day and night, the nighttime images have a significant reduction in image quality and detail. This is due to the camera's infrared capabilities, where grayscale images are collected when there is less light falling on the camera's light sensor. Similar visibility for a particular station can look drastically different if the image was collected during the day versus during the night (see Figure 3.4). In the night, certain landmarks can be obscured by the darkness and light sources that were previously not visible in the day become focal points in the image. Furthermore, the RGB and the grayscale images represent fundamentally two

different distributions of data. Due to the above concerns, I omit nighttime images from the data set and focus on estimating visibility for daytime images only (details in Appendix A.1).

In order to provide additional quality assurance to the daytime images, I filter the NYS Mesonet images by daytime or sunlight hours. Based on the latitude and longitude of the NYS Mesonet station, I use the associated sunrise and sunset time for a given day that an image was taken[1] (details in Appendix A.2). If the image's timestamp is outside the sunrise and sunset times, it is considered nighttime and if the timestamp is within the sunrise sunset times it is considered daytime. With this sunlight filtering, we guarantee that the RGB images gathered exclusively represent daytime scenes.

3.2.2 Image Defects and Data Continuity

Given that the NYS Mesonet is a large network, it is inevitable that there are occasional issues and inconsistencies with the collected data. These issues range from momentary occlusions to shifts in the camera position that affects years of data. One such temporary issue occurs during severe weather events when the camera lens may become blurred or occluded by particulate. This particulate could be from a number of sources such as snow, moisture, or leaves (Figure 3.5). However, when the camera occlusion is just moisture, one can typically still visually see important features in the scene for estimating visibility (Figure 3.6).

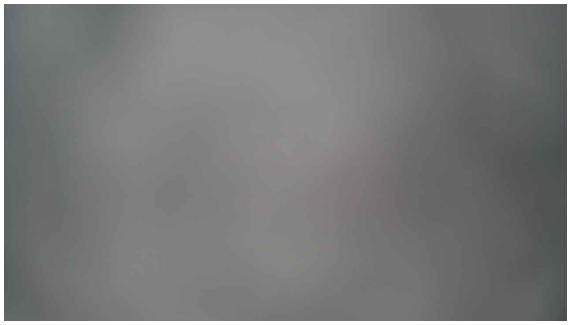
A more persistent issue is that the camera angle, orientation, or view may shift over time. This can be due to slight changes, such as the camera tilting or shifting, or can be more drastic, where the camera view is in a completely different direction than before.

[1] Used the Python package suntime (https://github.com/SatAgro/suntime)

(a) 4:15am EST (local time)

(b) 1:15pm EST (local time)

Figure 3.4: Images from the Potsdam station on May 21st, 2022. Image (a) was collected at 4:15am local time and image (b) was collected at 1:15pm local time. Both images have the same reported visibility of 10 miles.



(a)

(b)

Figure 3.5: Two occluded images from the Batavia station where the left depicts a total occlusion of impacted snow (a) and the right depicts a build up of water droplets or moisture on the lens (b).

When the camera's tilt changes slightly, the horizon line might not appear level across the image. When the camera has shifted, it can include more of the scene to the left or right, or perhaps include more of the ground or sky. Specifically the camera's position may drift over time, as demonstrated in Figure 3.7. At the Glen Falls station, there is a clear shift in the camera orientation, however, at the Potsdam station the camera position shifts only slightly. The different camera angles within the data set represent a pertinent variation to the data that is important for downstream deployment and usage, so they are not removed or rectified in the data set.

Figure 3.6: Image from the Batavia station with moisture build up on the lens circled in red.

However, the drastic changes to camera orientation are not acceptable in this context. Although uncommon, cameras may be re-positioned to avoid certain landmarks or to include other interesting scenery. This can prove difficult if we want fairly static scenes or landmarks in view. This violates the key assumption in the comparative approach, that the query and reference images should depict the same scene, just with different visibility conditions.

One example of this is the Manhattan site, where on June 21st 2018, the camera was positioned in a completely different direction (as shown in Figure 3.8). The previous position captured mostly the rooftop of a building. Whereas the new scene captures more of the city skyline and sky itself. For the Manhattan station, we exclude all images taken before June 22nd, 2018 in order to assure that all images from this station are relatively consistent in perceptive and composition.

Another example is illustrated by the Buffalo site where, in 2021, solar panels were installed at the location (Figure 3.9). The solar panels were installed over the course of 5 months from February to June of 2023. This dramatically changed the landscape of the Buffalo location perhaps impacting how visibility is characterized. However, the far tree line is maintained, so currently all years of data from this station remain in the data set.

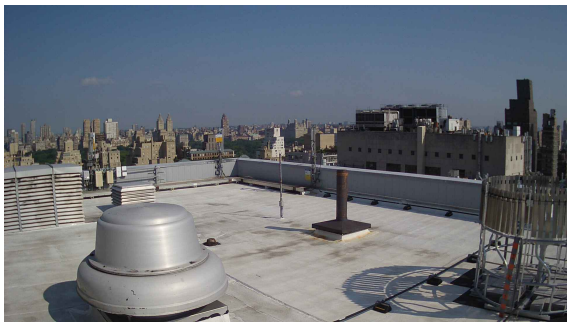
(a) Potsdam, July 2016

(b) Potsdam, July 2022

(c) Glen Falls, March 2017

(d) Glen Falls, July 2022

Figure 3.7: Images from the Potsdam and Glen Falls Mesonet stations where the left column is images taken during their first year established and right column is images taken from last year in the data set.



(a) Manhattan, July 2017

(b) Manhattan, July 2022

Figure 3.8: Manhattan site where (a) was the original camera view and (b) is after the re-positioning.

(a) Buffalo, March 2017

(b) Buffalo, June 2022

Figure 3.9: Buffalo station where (a) was collected in the first year of the stations deployment and (b) was collected after solar panels were installed.

3.2.3 Data Quantification

The first NYS Mesonet station was deployed in 2015. The standard network has grown incrementally and now consists of 126 unique stations. This means that different stations have collected a different number of images based on when the site was commissioned, amount of average light at the location, and different sunrise and sunset times. For this work I focus on ten specific stations in the NYS Mesonet Network. Justification for this decision is given in Section 3.3.

Each station has a unique deployment date. This impacts the total number of images collected for the first year of operation, and is reflected in our total image

counts per year by station, as shown in Table 3.1. Most stations collect approximately 65,000 images per year. In total, there are 4,064,673 images available for these 10 stations.

However, due to the concerns detailed in Section 3.2.1, I filter our data set to only include true daytime images as depicted by Figure 3.10. First, I filter by image color format and retain only the RGB images. Secondly, I filter by calculated sunrise and sunset times and retain all images that were taken after sunrise and before sunset. After filtering for just RGB images, the total number of images in the data sets is significantly reduced as seen in Table 3.2. The total count of available images drops from 4,064,673 to 3,516,075 (a 13% reduction in data set size). This reduction is mainly due to the one hour collection interval for grayscale images versus the 5 minute collection interval for RGB images.

However, if we look at how many grayscale images each station collects per year, as shown in Table 3.3, the Manhattan (MANH) and Queens (QUEE) stations rarely or never collect grayscale images. This is due to their hyper-urban location, where the scene is rarely dark enough to trigger grayscale image collection.

In order to truly filter out all the nighttime images, we can further filter the RGB images by extracting all images taken after sunrise and before sunset, as described in Section 3.2.1. The result of this process is shown in Table 3.4. The sunrise and sunset filtering reduces our overall data set size to 3,180,923. It is important to note that most stations collect around the same number of images each year, however, after filtering for just RGB images the Manhattan and the Queens stations have disproportionately more images because the grayscale mode rarely gets triggered due to the unique urban

	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
2022	69,043	67,298	68,056	69,322	69,322	69,378	68,970	68,578	68,686	68,851
2021	66,579	66,190	62,865	66,390	66,389	66,543	66,221	65,822	65,910	66,160
2020	65,435	64,832	64,298	63,004	64,828	65,334	53,317	65,327	65,670	65,812
2019	65,137	64,718	63,725	64,571	64,040	64,293	65,018	64,255	65,574	65,565
2018	65,004	64,955	64,492	65,015	64,288	65,015	34,484	63,864	66,238	66,046
2017	66,567	49,853	62,531	66,835	48,980	66,672	—	66,499	66,900	45,727
2016	58,854	—	32,891	63,159	—	49,111	—	41,086	28,281	—
Total	456,619	377,846	418,858	458,296	377,847	446,346	288,010	435,431	427,259	378,161

Table 3.1: image count by station and by year.

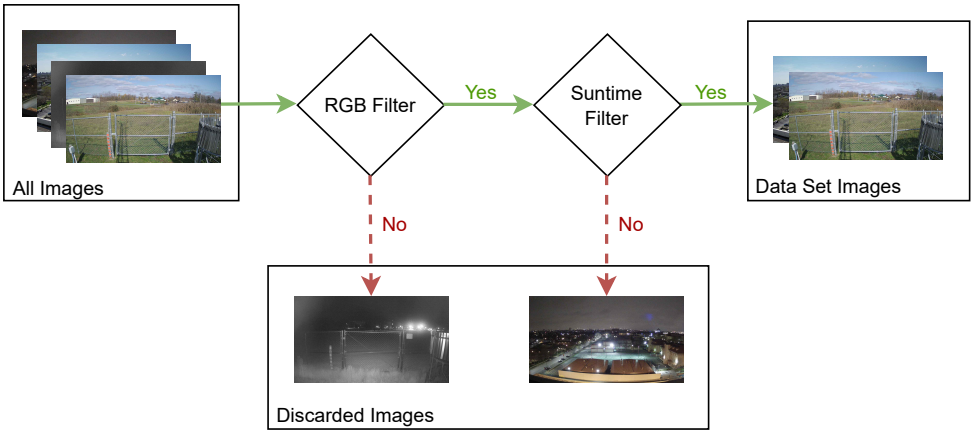


Figure 3.10: NYSM image processing steps.

lighting conditions. In contrast, the sunrise and sunset filtering restores the balance, where approximately the same number of images are available per station per year.

In summary, I filter the NYSM images to only include RGB images and images taken after sunrise and before sunset. I do this in order to guarantee I am only using daytime images for training, validating, and testing my models.

	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
2022	56,675	54,750	54,334	56,366	55,935	56,621	68,970	55,426	56,291	68,851
2021	56,528	55,903	52,978	56,207	55,488	56,337	66,221	54,865	55,918	66,160
2020	55,325	54,442	53,805	53,729	53,917	55,090	52,485	54,312	55,411	65,812
2019	55,188	54,434	53,026	54,491	53,456	54,192	65,018	53,634	55,492	65,565
2018	55,141	54,704	53,604	54,917	53,465	54,776	34,484	52,943	56,122	66,046
2017	56,741	42,568	52,900	56,729	41,539	56,592	—	56,252	56,816	45,704
2016	50,818	—	27,893	54,148	—	42,508	—	35,068	23,949	—
Total	386,416	316,801	348,540	386,587	313,800	376,116	287,178	362,500	359,999	378,138

Table 3.2: Image count for only RGB images separated by station and by year.

	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
2022	12,368	12,548	13,722	12,956	13,387	12,757	0	13,152	12,395	0
2021	10,051	10,287	9,887	10,183	10,901	10,206	0	10,957	9,992	0
2020	10,110	10,390	10,493	9,275	10,911	10,244	832	11,015	10,259	0
2019	9,949	10,284	10,699	10,080	10,584	10,101	0	10,621	10,082	0
2018	9,863	10,251	10,888	10,098	10,823	10,239	0	10,921	10,116	0
2017	9,826	7,285	9,631	10,106	7,441	10,080	—	10,247	10,084	23
2016	8,036	—	4,998	9,011	—	6,603	—	6,018	4,332	—
Total	70,203	61,045	70,318	71,709	64,047	70,230	832	72,931	67,260	23

Table 3.3: Image count for only grayscale images separated by station and by year.

	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
2022	52,212	50,547	51,275	52,299	52,539	52,687	52,837	52,030	52,096	52,825
2021	52,225	51,815	49,624	52,288	52,317	52,558	52,776	51,795	51,784	52,814
2020	51,049	50,418	50,382	49,916	50,783	51,301	42,239	51,171	51,271	52,362
2019	50,879	50,421	49,906	50,420	50,139	50,423	51,746	50,603	51,230	52,255
2018	50,844	50,684	50,555	50,906	50,409	51,119	27,343	50,044	51,850	52,703
2017	52,371	39,314	49,447	52,656	38,980	52,828	—	52,483	52,556	36,742
2016	46,893	—	25,858	50,238	—	39,535	—	32,244	22,064	—
Total	356,473	293,199	327,047	358,723	295,167	350,451	226,941	340,370	332,851	299,701

Table 3.4: Counts of images that are RGB and were collected between sunrise and sunset.

3.3 NYS Mesonet & ASOS Combined Data Sets

It is difficult to find image data sets that have corresponding visibility labels. Moreover, it is even harder to find data sets that are comprised of real-world images and visibility readings. For this work I build my own data set by integrating the NYS Mesonet camera images and the ASOS visibility measurements.

In order to assign ASOS visibility labels to the NYS Mesonet camera images, I pair up each of the NYS Mesonet sites to a nearby ASOS station. How "close" a NYS Mesonet site and ASOS station is determined by the distance and elevation distance between the locations. I compute the distance between two stations by using their longitude and latitude coordinates[2]. Both the NYS Mesonet and ASOS stations report their elevation, so I take the absolute value of the difference between the elevations to calculate the elevation difference. I pair up each NYS Mesonet site with its closest ASOS station by distance.

This works very well for some stations, such as the Glen Falls NYSM station and the Glen Falls/Warren County ASOS station. These two stations are less than a half mile away from each other, with an elevation difference of approximately 1 meter. This allows for a high correlation between the ASOS visibility measurements and the atmospheric visibility depicted in the NYS Mesonet images. However, for other stations, there is not a nearby ASOS station. For example, for the Roxbury NYSM station the closest ASOS station is approximately 45 miles away. In order to ensure that the pairs are reasonable I filter the station pairs such that they must be within 5 miles of one another and have an elevation difference of less than 70 meters. This limits the number of available NYS Mesonet stations from 126 to only 10 stations (see Table 3.5 and Figure 3.11).

NYSM Station (ID)	ASOS Station (ID)	Distance (mi)	Elevation Difference (m)
Batavia (BATA)	Batavia (GVQ)	1.81	0.67
Buffalo (BUFF)	Buffalo International Airport (BUF)	4.40	35.41
Elmira (ELMI)	Elmira/Corning (ELM)	4.42	45.32
Gabriels (GABR)	Saranac Lake/Adiron (SLK)	2.60	28.7
Glen Falls (GFAL)	Glen Falls/Warren (GFL)	0.42	1.34
Johnstown (JOHN)	Johnstown (NY0)	1.70	35.81
Manhattan (MANH)	New York City (NYC)	0.83	67.80
Penn Yan (PENN)	Penn Yan Airport (PEO)	3.58	47.33
Potsdam (POTS)	Potsdam (PTD)	1.88	11.78
Queens (QUEE)	New York/LaGuardia (LGA)	4.60	45.56

Table 3.5: NYS Mesonet and ASOS station pairs with distance and elevation difference information.

Once the stations are paired, I then assign a visibility label from the closest ASOS tower to each NYSM image. I do this by matching up the timestamps of the NYSM images to the timestamps of the ASOS visibility measurements. The collection time intervals of the NYS Mesonet and ASOS network are usually 5 minutes, however, the timestamps do not always align. When matching the timestamps, I allow for a tolerance of 2.5 minutes between the image timestamp and visibility timestamp. Without the tolerance, some stations would have little to no image and visibility matches. Table 3.6 shows the number of RGB images taken after sunrise and before sunset that had a corresponding visibility measurement.

In total, there are 2,075,107 daytime images with a corresponding visibility label. There are multiple reasons why this image and visibility measurement matching process reduces the overall image count. The NYS Mesonet stations collect daytime images at 5 minute intervals, and most the ASOS stations also collect their visibility measurements

[2]I used the Python GeoPy to calculate the distance between two latitude, longitude coordinates, (https://geopy.readthedocs.io/en/stable/).

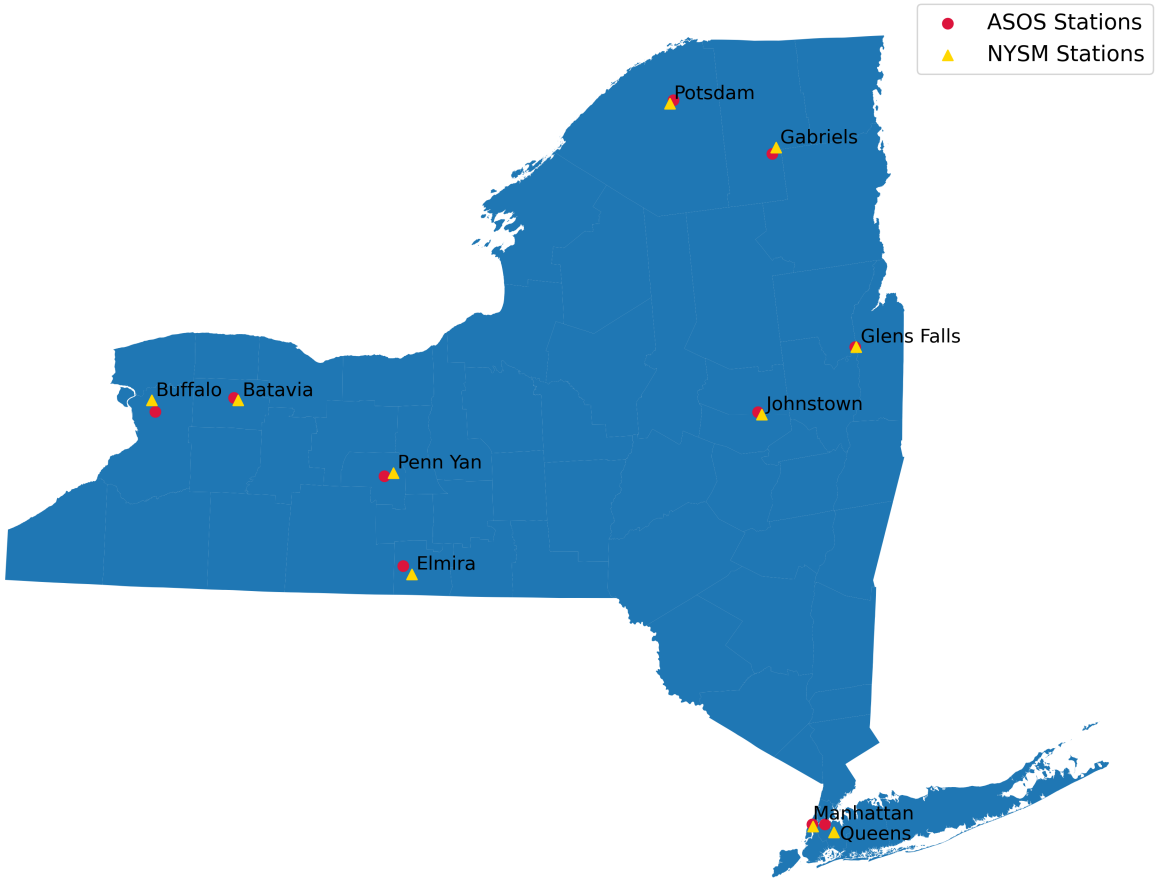


Figure 3.11: Active ASOS co-located with the nearest NYSM station, within 5 miles and a maximum elevation difference of 70 meters.

at 5 minute intervals. However, some ASOS stations have a limited number of years available, such as Batavia. The Batavia NYS Mesonet station has images from 2016 onward, but the Batavia ASOS station only had visibility readings from 2018 onward. In addition, some ASOS stations have a larger collection interval. The corresponding ASOS stations for Johnstown and Potsdam (NY0 and PTD) had visibility measurements for each 20 minute interval, and the ASOS station for Manhattan (NYC) had visibility measurements for each hour-long interval. Due to this, Johnstown, Potsdam, and Manhattan have the smallest number of images.

	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
2022	51,253	49,436	49,874	42,832	43,432	11,814	5,467	50,991	12,819	51,872
2021	51,731	47,908	48,357	42,591	39,418	11,619	5,481	50,407	12,860	51,867
2020	45,909	45,090	45,111	36,663	35,818	11,828	4,372	45,827	12,825	46,427
2019	48,783	50,018	49,455	40,927	39,876	9,438	5,512	50,149	12,836	51,500
2018	24,823	49,768	49,858	42,442	41,665	11,040	3,057	48,704	11,268	52,100
2017	—	38,553	47,182	42,738	21,244	4,932	—	47,948	—	35,852
2016	—	—	25,342	25,039	—	—	—	31,159	—	—
Total	222,499	280,773	315,179	273,232	221,453	60,671	23,889	325,185	62,608	289,618

Table 3.6: Data count of daytime image and visibility pairs separated by station and by year.

Chapter 4

Differential Approach to Estimating Visibility

Generalized absolute visibility prediction from a single image is an intrinsically hard problem. Low visibility conditions can be caused by various natural and artificial phenomena and look drastically different in the camera lens. The appearance of colors and lighting conditions can vary between locations, different weather conditions, time of day, or time of year. In addition, the distances between objects in the image and the camera are often unknown, and over time new structures can be built that change the composition of the scene. Other aspects of a scene may change over time, such as unexpected changes in camera orientation or routine changes due to the passing of the seasons. All of these factors complicate numerical visibility prediction from a single image, especially if a model has only been trained and evaluated on a specific set of visibility conditions and locations.

Past approaches focus on a single location or a small selection of unique locations, often focusing on identifying the presence or absence of specific landmarks in an image to inform the underlying visibility distance, e.g., Wauben and Roth (2016); Mi et al. (2020). Other approaches only evaluate their model on images from locations that the model has been trained to predict on, e.g., Palvanov and Cho (2019), Li et al. (2019b), and You et al. (2019). However, one can address the challenges of numeric visibility prediction by re-framing the problem. Instead of determining the numeric visibility distance from a single image, a model instead can learn to compare a pair

of two images from the same location to determine the relative visibility (You et al., 2019).

The goal of this work is to develop an approach that can generalize to new locations and sites without retraining the model. For my contributions, I propose a comparative visibility estimator that receives two images as input, a query and reference image, and predicts a probability that the reference image is more visible than the query image. In addition, the comparative visibility estimator can be extended to predict a numerical visibility value by producing enough comparisons between a query image and small set of reference images with known visibility distances. Furthermore, to guarantee cross-site generalizability, I construct my training, validation, and test sets carefully so they are temporally and spatially independent. With these design choices, the model cannot rely on knowledge of a specific location from training, nor are the predictions limited to one specific camera or site. This potentially allows the comparative visibility estimator to capture how fog impacts a scene more generally.

In this section, I describe the architecture of the comparative visibility estimator, the translation between comparative visibility estimates to a numerical visibility prediction, and the careful partitioning of the combined ASOS and NYSM data set to guarantee cross-site generalizability.

4.1 Comparative Visibility Estimator Architecture

Relative visibility estimation has shown to be a successful approach (You et al., 2019). However, previous work requires large quantities of hand curated and labeled data, and was not tested on novel locations. Furthermore, many approaches require large pre-trained models with millions of parameters, such as in Li et al. (2017, 2019b, 2021).

In my approach, I use a CNN-based model with a moderate number of parameters to predict the comparative visibility between two images.

The comparative visibility estimator has fewer than 1 million trainable parameters (808,065). The model, shown in Figure 4.1, receives two images as input, the query image and the reference image. In deployment, the query image is a live image from a site with an unknown visibility distance while the reference image is a historical image of the site with a known visibility distance. These images are resized to (256, 256, 3) and concatenated along the channel dimension in order to be passed as input into the model.

The model itself consists of two parts, the CNN block and the dense neural network (DNN) block. A DNN is an artificial neural network (ANN) built using multiple dense layers. A dense layer is composed of neurons that receive their input from all the neurons in the preceding layer. The CNN block of this model extracts spatial features about the inputs, while DNNs ignore spatial properties between the features. In this model, the CNN block works as a spatial pattern detector by extracting relevant features related to visibility. The DNN block determines the model output based on the synthesized features. The CNN block consists of several layered, repeated modules of convolution, spatial dropout, and batch normalization (see Figure 4.2). The convolutional layers are a 2D spatial convolution where learned filters are applied over the input tensors with different filters to extract hierarchical features in the input. Spatial dropout is a form of regularization that can be added to convolutional layers, where entire features are randomly removed from the network during training in order to encourage redundancies between the feature maps and prevent over-fitting to the training set. Batch normalization is a technique that standardizes the inputs to each of the layers by re-centering and re-scaling the input with respect to the input's mean

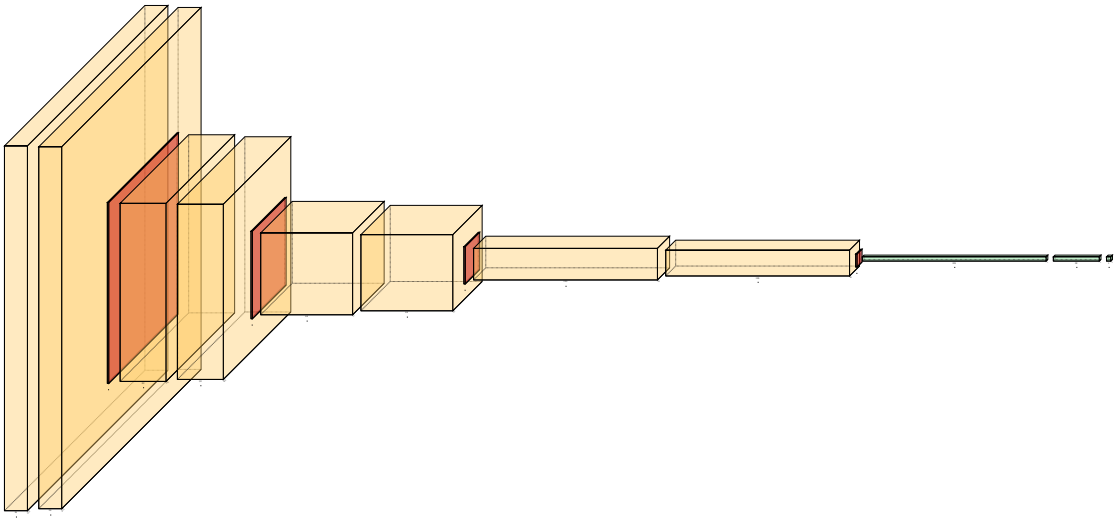


Figure 4.1: The comparative visibility estimator architecture where the yellow represents the CNN modules, and the green represents the DNN layers (Iqbal, 2018).

and standard deviation. Every two CNN modules are followed by a layer of max pooling. Max pooling is a method of down-sampling between layers, which allows subsequent convolutions to have an effectively wider receptive field. Max pooling also allows for models to run faster since intermediate representations are smaller, but also can improve input translation invariance to the model, since small vertical or horizontal shifts in the inputs do not affect the pooling outputs. Figure 4.2 shows one CNN module that makes up the CNN block.

The final output of the CNN block is condensed into a feature vector of size (128,) by using global max pooling. Global max pooling is similar to max pooling, however, the maximum value is collected across the space of an entire channel. This step effectively asks the question of whether each high level feature exists within the input images. The output feature vector is then fed into the DNN portion of the model, which reasons about the combined existence of features in the input images in order to make the

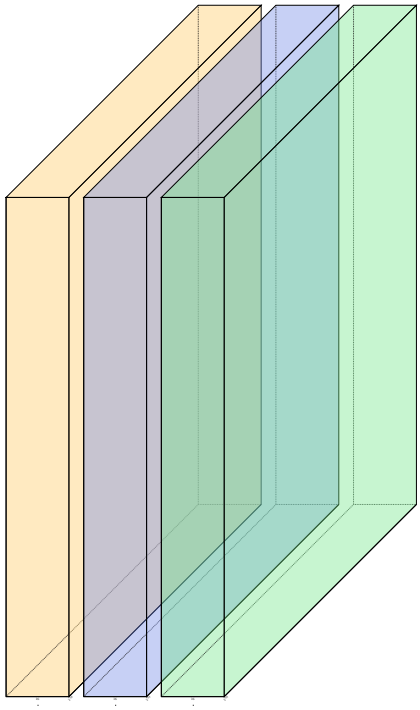


Figure 4.2: One CNN module within the comparative visibility estimator, where yellow represents the convolutional layer, blue represents the spatial dropout layer, and green represents the batch normalization layer (Iqbal, 2018).

final prediction. The DNN block consists of two dense layers with dropout and a final output layer consisting of a single neuron. The overall model can be see in Figure 4.1.

The output neuron has a sigmoidal activation function and represents the probability that the query image visibility is less than the reference image visibility. This is also written as $Pr(Q < R)$, where Q is the query image's true visibility in miles and the R is the true reference image's visibility in miles. Furthermore, when $Pr(Q < R) = 1$, then the query image is interpreted as less visible than the reference image (the query image has more fog or obscurities); when $Pr(Q < R) = 0$, then the query image is interpreted to be more visible than the reference image (the query image has less fog or fewer obscurities), and finally when $Pr(Q < R) = 0.5$ the query image and the reference image are interpreted to have the same visibility. These interpretations are

invariant to the underlying range of visibility, thus as $Pr(Q < R) = 0.5$ could be true of a query and reference image pair where both have a visibility of 0.13 miles, or where both have a visibility of 10 miles. The model was trained using binary cross-entropy, additional training details are included in Appendix A.3.

4.2 Computing Estimates of Absolute Visibility

The comparative visibility estimator only assesses the relative visibility of two images, not the numerical visibility of a unknown query image. However, with a small sample of labeled reference images, the comparative visibility estimator can produce output probabilities between a single query image and the set of reference images in order to infer a numerical visibility estimate. This is done in two steps: 1) identifying an accurate set of reference images, and 2) translating the comparative visibility estimator's output probabilities between a novel query image and the set of reference images to a numerical visibility estimate.

4.2.1 Reference Image Selection

Reference image selection is crucial to the success of absolute visibility estimation. If a reference image is incorrectly labeled, the predicted visibility can be greatly skewed. Reference image selection can be done by hand with an expert selecting examples for each visibility distance, or instead can be done using a self-supervised inspired method. For this work, the reference images were selected in an automated self-supervised way. After the comparative visibility estimator is trained, images that often produced correct predictions when used as reference images were selected to be candidates for the reference image set.

The reference image selection process happens after the comparative visibility estimator has been trained. Candidate reference images were only selected from the training sets in order to preserve the independent nature of the validation and test sets. In addition, reference images were only selected from images that were collected in 2019 in order to speed up the process.

For each station, a subset of the images are randomly selected from each visibility distance bin. This creates a subset over the entire year that is balanced, approximately representing each unique visibility distance equally. Collecting a balanced subset is necessary in order to reduce the number of total possible image pairs within this set to a computationally feasible number. Each image in the subset is then compared to all the other subset images using the trained comparative visibility estimator. Each image has its accuracy over all the comparisons recorded. The images are then ranked with respect to their accuracy. For each unique visibility distance, the five images with the highest accuracies are saved as reference images. This produces a reference image set composed of five images per unique visibility distance bins.

For example, a balanced subset was created from images collected at the Batavia station in 2019. The subset images with reported visibility distance of 0.13 were compared to all other subset images which had visibility distances ranging from 0.13 to 10 miles. For a particular image, if all comparisons containing this image were accurate, the image would be considered to be an ideal reference image. The top five images with the highest overall accuracy are selected for each visibility distance. In this case, Batavia had 19 unique visibility distances represented in 2019 and 5 images were selected for each unique distance. So, 95 reference images in total were selected to be part of Batavia's reference image set.

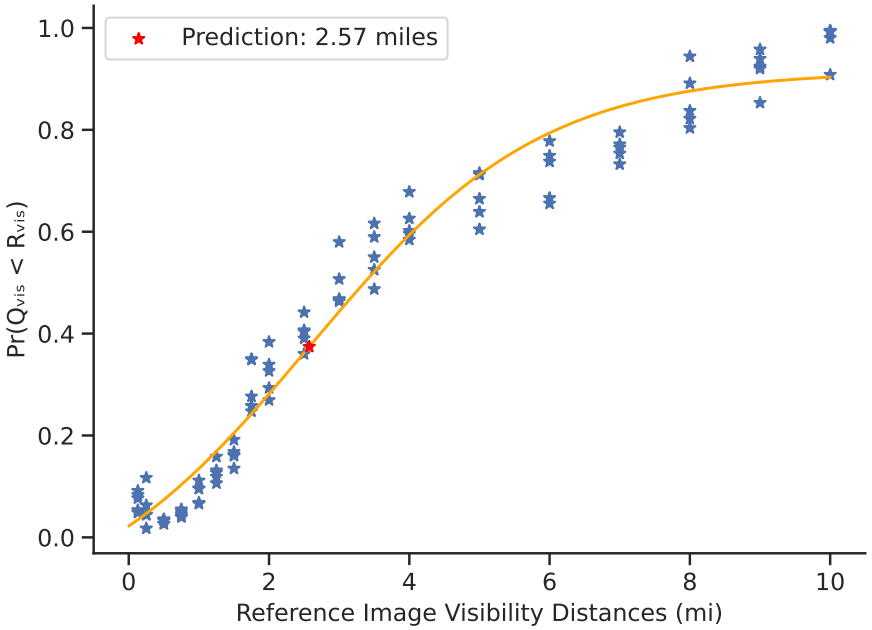


Figure 4.3: Example sigmoidal fit for model output probabilities.

4.2.2 Translating Comparisons to Numeric Visibility Distance

The comparative visibility estimator only predicts the probability of the query image being less visible than the reference image. However, with a labeled set of reference images, one can translate the output probabilities of the comparative model into a numerical visibility prediction for a novel query image. The output probabilities of the model with respect to reference image visibility should follow a sigmoid shape as the reference images transition from clearly less visible to clearly more visible than the query image.

Figure 4.3 depicts the comparative model's output probabilities between a singular query image with a known visibility distance of 2.5 miles and a set of reference images with visibility distances ranging from 0.13 to 10 miles, where there are five reference images for each unique visibility distance. Each blue star is the output probability from the comparative model between the query image and the corresponding image

from the reference set. The output probabilities from the model exhibit an "S" like shape. I use a sigmoidal function to model this relationship:

$$y = \frac{L}{1 + e^{-k(x-x_0)}} + b \ , \tag{4.1}$$

where L is the amplitude, x_0 is the x-value of the inflection point, k is the gain, and b is the bias term. I find the optimal values of these parameters to fit the sigmoidal function[1] to the output probabilities (Pedregosa et al., 2011). Ideally, the output probabilities from the comparative model should be closer to 0 when the reference image visibility is low. Moreover, the output probabilities from the comparative model should closer to 1 when the reference image visibility is high. All of the images in the data set have a visibility label greater than 0 and less than 11. So, I add bias points (-10, 0), (0, 0), (11, 1), and (20, 1) to bias the fitted sigmoid towards a solution that reflects the expected behavior, especially along the edges of the graph. Furthermore, I restrict the gain term, k, to be positive; this guarantees that the fitted curve is an increasing sigmoidal function. I also restrict x_0 to be within the standard visibility distances in the data set, 0.13 to 10.0 miles.

I use the x-value of the inflection point, x_0, as the numerical visibility prediction. Ideally the amplitude, L, should be 1 and the bias term, b, should be 0, however, this often not true for the curve of best fit. Although a probability of 0.5 from the comparative model can signify the equivalence case or where the model is unsure, a point along the fitted curve that holds a y-value of 0.5 may be closer to either asymptote of the sigmoid. The inflection point is surrounded by the points where the transition from 0 to 1 is at its steepest. Furthermore, inflection point is equidistant from each

[1] I used curve_fit() from the Python package scikit-learn to fit the sigmoidal function to the comparative probabilities. I use the default number of maximum iterations of 800.

of the sigmoid's asymptotes. The steep transition between 0 and 1 pins the visibility prediction between the asymptotes.

The inflection point was found to more accurate for numerical visibility prediction than using the point whose y-value was closest to 0.5. This was confirmed through informal empirical testing. A example of this is shown in Figure 4.3. The curve's x_0 was fit to be approximately 2.57, which is within 0.07 miles of the true visibility of the query image (2.5 miles). However, the prediction for the query image would be greater than 3 miles when using the point on the fitted curve with a y-value closest to 0.5.

4.3 Data Set Procedures for Generalization

In order to produce models that are generalizable to new sites, the data must be carefully structured and handled to avoid autocorrelation between the training, validation, and test sets. Like most real world data sets, my combined NYSM and ASOS image-based visibility data set requires a certain amount of data cleaning. Furthermore, visibility data sets tend be quite imbalanced since low visibility conditions are far less common than high visibility conditions. For these reasons, I implement a detailed data handling pipeline with data set procedures for preventing autocorrelation between the sets, data cleaning and standardization, as well as data balancing. I use a version of cross validation, where my folds are temporally and spatially independent, to evaluate the cross-site generalizability of my models. In addition, I clean and standardize the ASOS visibility labels and speed up training by resizing the NYSM images. I also implement multiple versions of data balancing in order to assure my model performs well in all visibility conditions for multiple stations.

4.3.1 Orthogonalized Cross Validation

Due to the difficulty of producing robust visibility data sets and the challenge of cross-site and generalizable visibility prediction, there has been limited to no work on demonstrating a visibility approach that is truly generalizable to scenes not represented in the training set. An integral goal of this work is to create a general visibility estimator that can be applied to any location. Since we have a small number of stations, one important step to ensure cross-site generalizability is to perform cross validation with spatially and temporally independent folds.

A fold is a specific partition or subset of the data. For this work, each fold is a single year of data for a single station. Multiple folds make up a single rotation of the data. Furthermore a single rotation is composed of folds from eight stations for the training set, a fold for the validation set, and a fold for the test set. For cross validation purposes, we iterate through 10 different rotations of the data set, where each rotation is composed of a different subset of the folds.

For temporal independence, I break up the folds by calendar year. Since different stations were established during different years, all years up to and including 2020 are used for the training sets. Furthermore, 2021 is reserved for validation sets and 2022 is reserved for test sets. Partitioning the data sets by year is a vital step that breaks the temporal autocorrelation between the training, validation, and test sets. Low visibility events are largely underrepresented in many data sets. This is because sunny or clear conditions are much more common than foggy or severe weather conditions. If there is a severe storm that impacted visibility for an extended period of time, it is important to keep the examples from that incident all in one partition to reduce auto-correlation between the sets.

For spatial independence, I also break up the folds by station. Eight stations are included in the training set, one station is reserved for the validation set, and one station is reserved for the test set for each rotation. As I iterate through the rotations, I rotate to the right creating a new subset of stations for the training, validation and test sets (see Table 4.1). In rotation 0, Postdam 2021 and Queens 2022 are reserved for validation and test sets, respectively. In rotation 1, the training, validation, and test sets shift to the right by one station, making Queens 2021 and Batavia 2022 the validation and test sets, respectively. This process is performed 10 times, so that each station becomes a validation set exactly one time and a test set exactly one time. This means there is overlap between the training sets across the different rotations. However, the validation sets are completely independent as they each have a unique station and year. This is also true for the test sets.

4.3.2 Data Standardization

Although seemingly continuous, the ASOS readings are reported in specific statute mile increments. The specific statue mile increments are not standard for all stations. Some stations' lowest possible reported visibility is 0.06 mile; for others it is 0.25 mile. I use the standard visibility distances that are reported by ASOS of 0.13, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5, 4, 5, 6, 7, 8, 9, and 10 miles. This is the same set of standard increments found in the ASOS User's Guide (National Oceanic and Atmospheric Administration, 1998), with the exception of 3.5 miles. I add 3.5 miles because it is commonly found in the available ASOS visibility data. If a visibility reading in the data set is found to be outside of the standard visibility distances, I replace the offending visibility reading with the closest standard visibility distance. One common example where this takes place is when the visibility distance is reported as 0.24 miles. I group the 0.24 mile samples into the 0.25 visibility distance bin.

In addition, I resize the NYSM images. The NYSM images originally have a height of 720 pixels and a width of 1280 pixels. Originally, I used full size images for training, however, training often took many days at this resolution, so I resize the images to a square (256 x 256) to improve training time. Resizing to a square resolution also decreased training time without affecting performance. For all my results shown I use an image size of 256 x 256.

4.3.3 Balanced Sampling

Low visibility conditions are largely underrepresented in many data sets. This is also true for the combined NYSM and ASOS data set. In addition, some stations have more years of data available than other stations. In order to prevent over-fitting to highly visible examples or specific camera locations, I implement two levels of data set balancing and sampling used in the training and validation sets.

I first address the imbalance in visibility readings. For each station, I equally sample images from each of the standard visibility increments. Figure 4.4 depicts this process, where all the data for Batavia is split into the different visibility distance bins, then the data is equally sampled from each of these bins. This means that the probability of retrieving an image with a visibility of 0.13 miles is the same as retrieving an image with a visibility of 10.0 miles. This means that over the course of training certain images will be sampled more often than others.

After sampling from the visibility distance bins and pre-processing the data, I address the imbalance of data between stations. In this case, I sample images from each of the stations with equal probability (see Figure 4.5). This means specific images from stations like Manhattan are repeated more often than images from a station like Batavia. I implement these sampling techniques using the TensorFlow Dataset class,

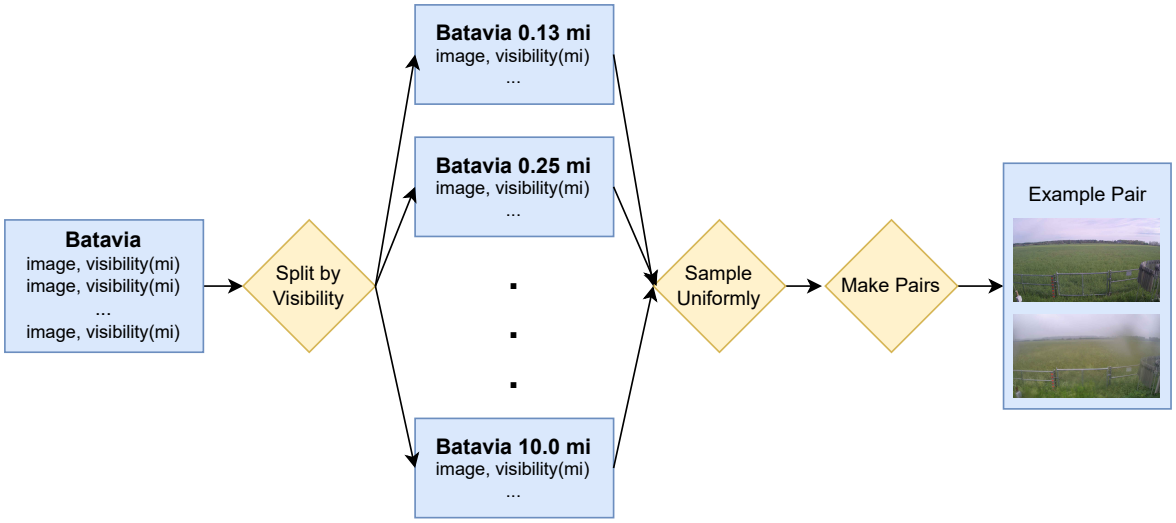


Figure 4.4: Steps for sampling across different visibility distances for a single station, where the data is separated by visibility distance bin. When an example is requested, images are uniformly sampled across the different visibility distance bins.

where I create individual Dataset objects for each station, and nested data set objects for each of the visibility increments within the station.

	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
2022	51,253	49,436	49,874	42,832	43,432	11,814	5,467	50,991	12,819	51,872
2021	51,731	47,908	48,357	42,591	39,418	11,619	5,481	50,407	12,860	51,867
2020	45,909	45,090	45,111	36,663	35,818	11,828	4,372	45,827	12,825	46,427
2019	48,783	50,018	49,455	40,927	39,876	9,438	5,512	50,149	12,836	51,500
2018	24,823	49,768	49,858	42,442	41,665	11,040	3,057	48,704	11,268	52,100
2017	—	38,553	47,182	42,738	21,244	4,932	—	47,948	—	35,852
2016	—	—	25,342	25,039	—	—	—	31,159	—	—

	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
2022	51,253	49,436	49,874	42,832	43,432	11,814	5,467	50,991	12,819	51,872
2021	51,731	47,908	48,357	42,591	39,418	11,619	5,481	50,407	12,860	51,867
2020	45,909	45,090	45,111	36,663	35,818	11,828	4,372	45,827	12,825	46,427
2019	48,783	50,018	49,455	40,927	39,876	9,438	5,512	50,149	12,836	51,500
2018	24,823	49,768	49,858	42,442	41,665	11,040	3,057	48,704	11,268	52,100
2017	—	38,553	47,182	42,738	21,244	4,932	—	47,948	—	35,852
2016	—	—	25,342	25,039	—	—	—	31,159	—	—

	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
2022	51,253	49,436	49,874	42,832	43,432	11,814	5,467	50,991	12,819	51,872
2021	51,731	47,908	48,357	42,591	39,418	11,619	5,481	50,407	12,860	51,867
2020	45,909	45,090	45,111	36,663	35,818	11,828	4,372	45,827	12,825	46,427
2019	48,783	50,018	49,455	40,927	39,876	9,438	5,512	50,149	12,836	51,500
2018	24,823	49,768	49,858	42,442	41,665	11,040	3,057	48,704	11,268	52,100
2017	—	38,553	47,182	42,738	21,244	4,932	—	47,948	—	35,852
2016	—	—	25,342	25,039	—	—	—	31,159	—	—

Table 4.1: First three out of ten rotations in the orthogonalized cross-validation procedure, where the years are depicted along the left-most column and stations are depicted along the top-most row. Each cell is a single year of data for a single station, or otherwise known as a fold. The green highlighted cells are the training sets which are composed of multiple folds or ”cuts” of data, the blue highlighted cells are the validation sets which are composed of a single fold for each rotation, and the red highlight cells are the test sets which are composed of a single fold for each rotation.

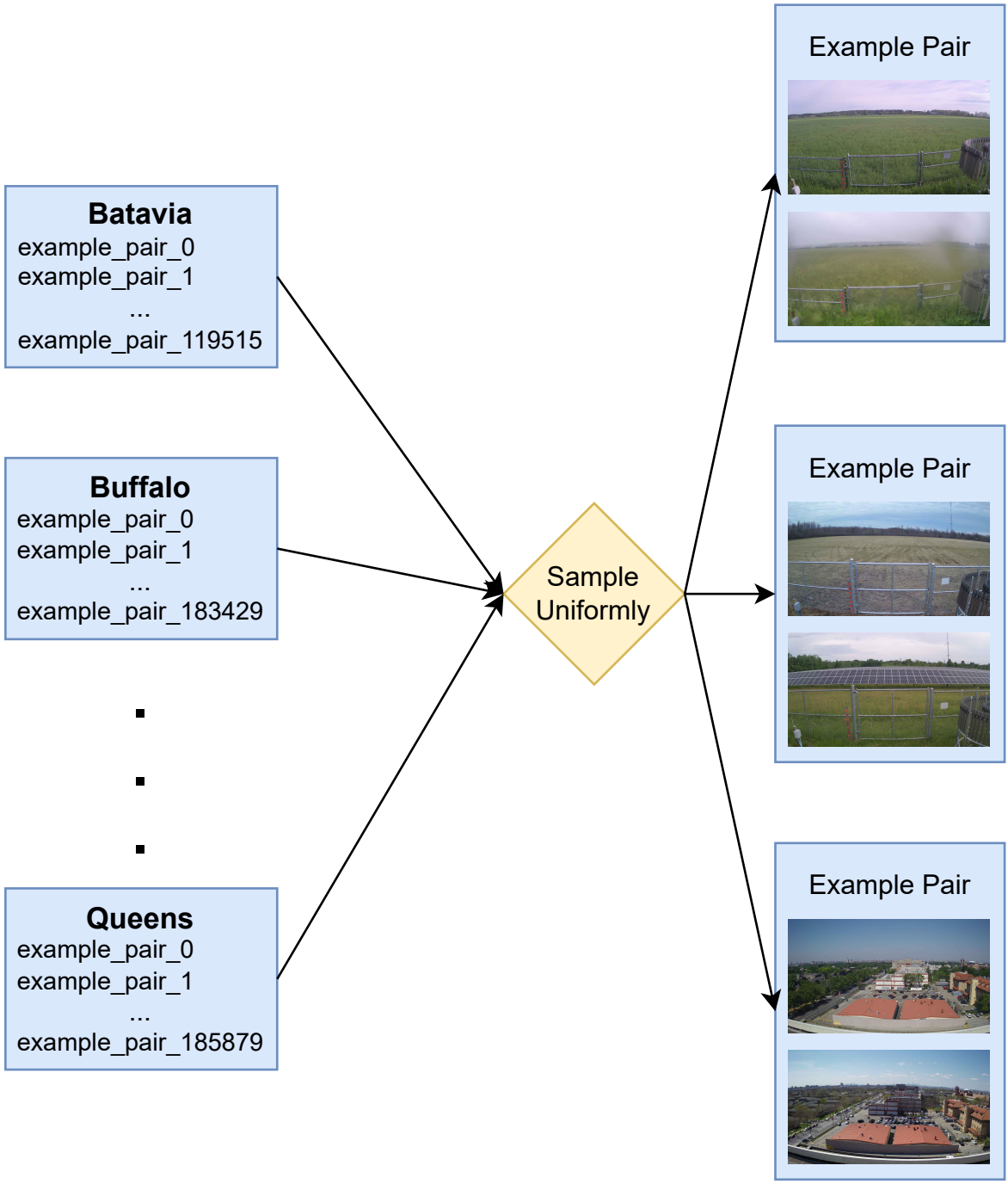


Figure 4.5: Steps for sampling across different stations, where each station's data is already pre-processed and image examples are paired up. When an example is requested from this pipeline, images are uniformly sampled across the different stations.

Chapter 5

Results

5.1 Comparative Model Results

There are three possible interpretations for outputs for the model; query image is less visible than reference, query has the same visibility as reference (or the equivalence case), or query image is more visible than reference. However, when computing accuracy it is unclear how close to 0.5 the model output has to be in order to predict the equivalence case. For the purposes of presenting comparative results and computing binary accuracy, the model's output represents $\Pr(Q \leq R)$, where if the query and reference images are equal and model returns 0.5 or above, the prediction is considered correct when computing accuracy. This is does mean that if the query and reference images are equal and model's output is 0.49, the prediction is considered mis-classified when computing accuracy. This is just for computing binary accuracy, and did not impact how the models were trained.

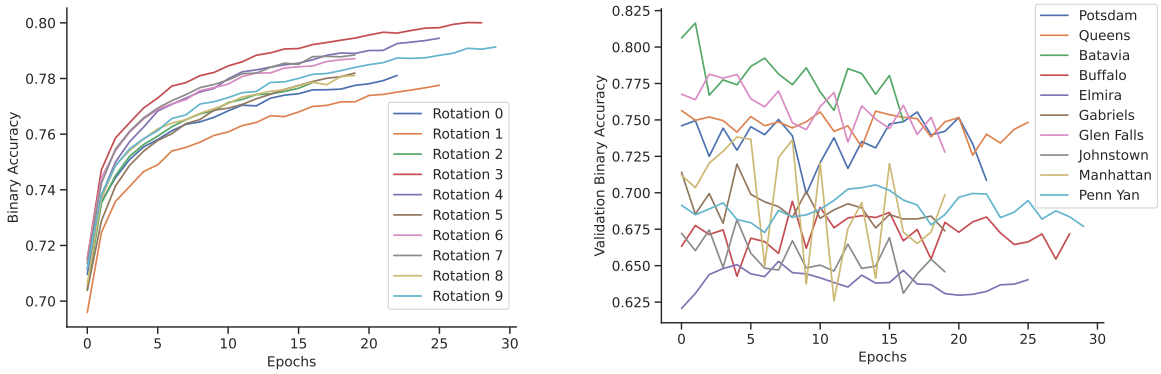
Figure 5.1 shows the training and validation set accuracy learning curves. During training, the accuracy of the training sets converged to an accuracy of over 77% (Figure 5.1) and the average validation accuracy during the epoch with the lowest loss is approximately 72.7%. However, performance on the validation sets quickly reached its maximum within the first few epochs of training. Since each validation set is composed of only one station, the validation performance greatly reflects the difficulty and unique

characteristics of each of the stations. In addition, the training results vary widely between each station. In this section, I will breakdown the results for the comparative visibility estimator based on station and year.

5.1.1 Evaluation Procedures and Under-sampling

When evaluating the results of the model, it is not feasible to produce results for every possible pair in the training, validation, or test sets. For example, the station Batavia has 50,879 images from the year 2019 and if every possible pair was considered, the model would have to evaluate $50,879^2$ image pairs. Unfortunately, this would be computationally infeasible. Furthermore, due to the label imbalance in the data set, most of those pairs would composed of high visibility examples. In order to efficiently and appropriately assess model performance, I create a subset of the data for evaluation purposes.

For under-sampling the data, I first find the visibility distance that is least represented in the data set. Let k be the sample size for under-sampling. The sample size, k, is initially set to be the number of images from the least represented visibility distance. The sample size k is then bounded between 30 and 100, inclusive. This is to make sure that there are enough images for an effective evaluation, but not too many images, so that evaluation procedures terminate in a reasonable amount of time. This means that if the number of images from the least represented visibility distance is less than 30, k is set to be 30. If the number of images from the least represented visibility distance is greater than 100, k is capped at 100. I sample without replacement k images from each of the visibility distance bins. In the case that there are less than 30 images in a visibility distance bin, all images from that visibility bin are sampled.



(a) Training accuracy

(b) Validation accuracy

Figure 5.1: Training and validation accuracy for each of the rotations.

For example, in 2018, the station Penn Yan (PENN) only had 50 images with a reported visibility of 0.13 miles, which was the least represented visibility distance. I sampled all 50 images with a visibility distance of 0.13 miles, only 50 images with a visibility distance of 0.25, only 50 images with a visibility distance of 0.50, and so on. In 2021, Penn Yan (PENN) only had 10 images with a reported visibility of 0.13 miles. I sampled all 10 images with a visibility distance of 0.13, then sampled 30 images with a visibility distance of 0.25, then sampled 30 images with a visibility distance of 0.50, and so on. In 2019, Penn Yan (PENN) had 167 images with a reported visibility of 0.50 miles. For this case I sampled 100 images for each of the visibility distances in the data set. This sets the upper bound on the number of possible pairs to be $1,900^2$ or 3,610,000 pairs per year per station.

5.1.2 Training Results

Although there are many years of data included in the training sets, there is little difference in performance between different years in the training set. For brevity, I only show performance for 2019 in the following training results. I selected 2019 because in

my original experiments 2019 was the most current year in the training set. However, in this work I added 2022 to the data set, but retained 2019 as the year I used to primarily evaluate the training set.

As shown in Figure 5.1 b. the model performed better on some stations and worse on others. An example of a high-performing station is Batavia. Figure 5.2 shows the accuracy of the under-sampled set broken down by query and reference visibility distances. Each square is the accuracy for comparisons between images with a specific query visibility distance and a specific reference visibility distance. For example, for all pairs between query images with a visibility distance of 2.5 miles and reference images with a visibility distance of 1.0 miles, the model achieved an accuracy of 92%.

The upper-left and lower-right corners represent the pairs that have the highest difference in visibility. As expected, the model performs best on these cases since there is a clear visual distinction between 0.13 miles and 10.0 miles. However, near the diagonal, the difference in visibility is less clear. When the query image has a visibility distance of 2.0 and the reference image has a visibility distance of 1.75, the model only has a 55% accuracy. The model ability to distinguish between 2.0 and 1.75 is essentially a coin flip, however, this is not necessarily a flaw in the model itself. Although there may be some subtle differences between 1.75 and 2.0 miles of visibility, there may not be substantial visual differences in the image to represent the visibility difference. For example, in Figure 5.3, the image pair technically has a visibility difference of 0.25 miles. The model should predict 0, since the query image label has a greater visibility distance than the reference image. However, in this case, the model predicted a probability of 0.61 which is closest to the equivalence case.

There is some asymmetry in many of the accuracy grids, as seen in Figures 5.2 (see Appendix B.1 for all accuracy grids). For the Batavia station, this mainly occurs in

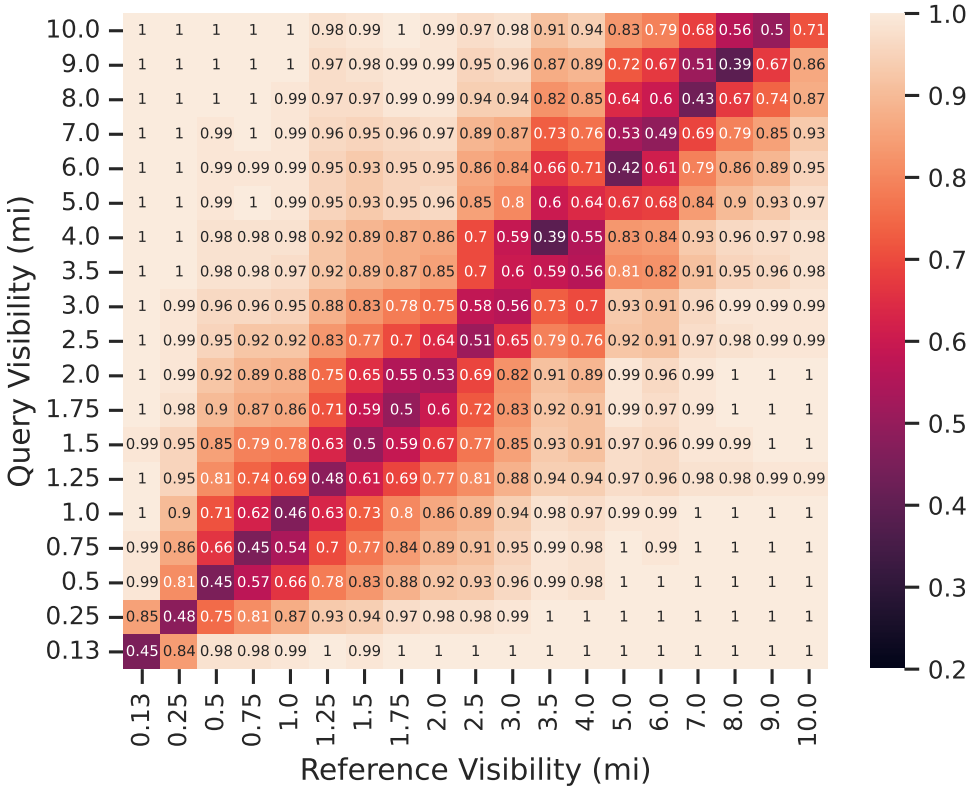


Figure 5.2: Accuracy grid for Batavia 2019 images for rotation 0.

the upper-right hand corner, where it becomes difficult to distinguish between visibility distances in the 5.0 to 10.0 mile range. This may be due to the image composition itself. Although ASOS visibility sensors report visibility distances up to a maximum of 10 miles, many stations do not have a field of view that encompasses landmarks that are 10 miles away, and often do not even have landmarks that are 5 miles away. For Batavia, the main visual feature is a tree line that cuts half way across the image (Figure 5.5a). This tree line is between 0.25-0.5 miles away from the camera itself. The main feature that give an indication of visibility is how visible the tree line is or how cloudy the sky is. Although, fog is easily spotted in the scene for lower visibility conditions, such as 0.13, 0.25, 0.5 miles, the appearance of the scene does not change significantly in the

(a) Query image, 2.0 miles

(b) Reference image, 1.75 miles

Figure 5.3: Example from Batavia 2019 where the query image (a) has a visibility distance of 2.0 miles and the reference image (b) has a visibility distance of 1.75 miles. The model's prediction for this pair was 0.61.

high visibility cases. This makes it more difficult or nearly impossible to distinguish between high visibility conditions such as 8, 9, or 10 miles.

On the diagonal, the query and reference images have the same visibility distance and the model should predict close to 0.5. When translating an output probability to a binary class the default threshold is 0.5. So for all examples along the diagonal the expected accuracy is approximately 50%, where half the time the model will overshoot above 0.5 or undershoot below 0.5. In rotation 0, the overall accuracy for Batavia in 2019 was 87.1%, however, if we exclude the equivalence cases the accuracy becomes 88.9%.

Batavia is a station with a wide view of the landscape and a clear view of the sky. In addition the NYS Mesonet camera and ASOS visibility sensor are relatively close where the distance between the camera and sensor is 1.81 miles and the an elevation difference is 0.67 meters. Glen Falls is another higher performing station where the camera and the ASOS sensor are in close proximity (0.42 miles in distance and 1.34 meters in elevation difference). In rotation 4, Glen Falls has an overall accuracy of 83.1% and an accuracy of 85.1% when excluding equivalence cases. It has similar performance

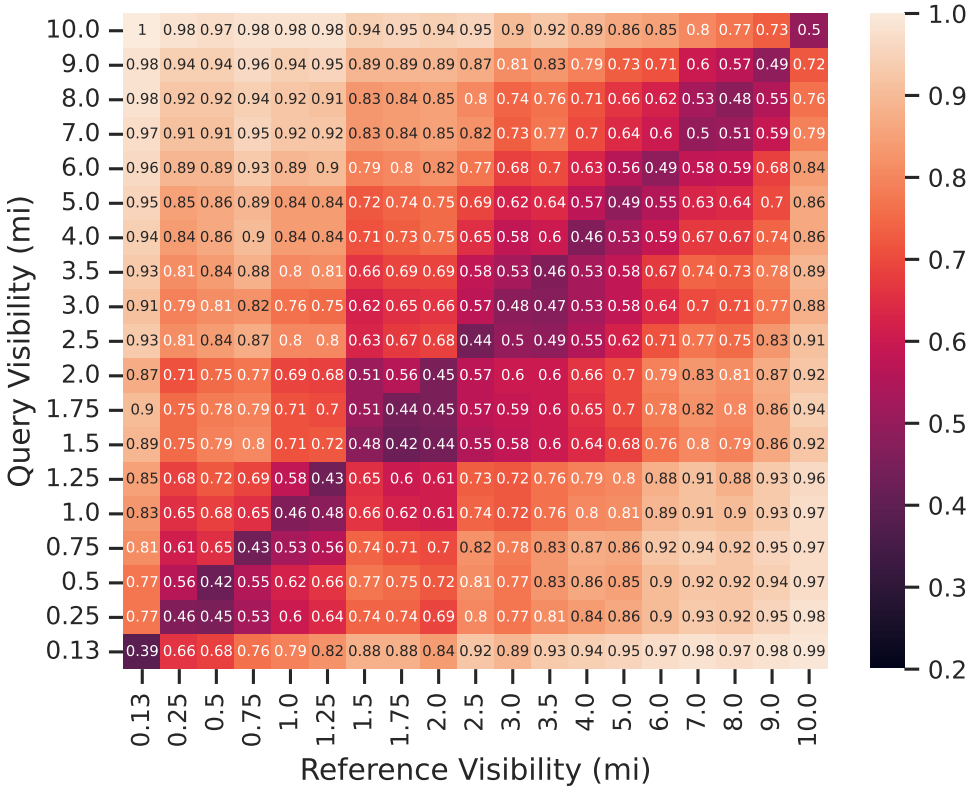


Figure 5.4: Accuracy grid for Buffalo 2019 images for rotation 1.

to Batavia with some asymmetry and low accuracy along the diagonal. A complete description of training accuracies over the years and folds is in the Appendix B.1.

Not all stations have the same attributes and this is reflected in the training performance. At the Buffalo station, the model was able to capture the differences between high and low visibility distances, however, when it came to closer visibility distances such as 1 and 2 miles, the model was unable to differentiate between the visibility conditions (see Figure 5.4). The Elmira station had similar issues, with overall low accuracy between the different visibility pairs (Figure Appendix B.3a).

In rotation 1, the Buffalo station received an overall accuracy of 75.4% and an accuracy of 77.0% excluding equivalence cases. In rotation 2, the Elmira station received and overall accuracy of 72.4% and an accuracy of 73.9% excluding equivalence cases.

(a) Batavia

(b) Queens

(c) Buffalo

(d) Elmira

Figure 5.5: Clear day images of (a) Batavia, visible tree line is between 0.25 and 0.5 miles from camera, (b) Queens, city line buildings are approximately 8-9 miles from camera, (c) Buffalo and (d) Elmira, tree lines are less than 0.13 miles from camera.

The Buffalo and Elmira stations suffer from similar issues. The NYS Mesonet camera and ASOS visibility sensors for Buffalo and Elmira are over 4.4 miles away from each other for both sites, and have an elevation difference of 35 and 45 meters between the sensors, respectively. The distance between the camera and the ASOS sensor adds to a de-correlation between the visibility phenomena in frame and what is captured by the visibility sensor.

However, this was not the only problem that impacted the results. For both stations, the camera has a short field of view with limited access to broad sections of sky (Figures 5.5c and 5.5d). One station that that has similar sensor limitations, but provided results comparable to Batavia and Glen Falls is the Queens station. In rotation 9, Queens also performed well with an overall accuracy of 83.2% and an accuracy

of 84.9% excluding equivalence cases. The NYS Mesonet camera and ASOS visibility sensor for the Queens station are 4.60 miles in distance and an elevation difference of 45.56 meters. However, the image shows a broad and lengthy view of the scene where some visible buildings and landmarks are 8-9 miles away from the camera(see Figure 5.5b). All accuracy grids for the training set are included in Appendix B.1.

5.1.3 Validation and Test Results

For all stations, the year 2021 was reserved for validation and the year 2022 was reserved for testing. This means the validation set for each rotation is comprised of examples from a single site that was not represented in the training set, and a completely novel year of data. The same is true for each of the test sets.

Accuracy results for each of the validation and test sets are presented in Table 5.1 and Table 5.2. In Table 5.1, the overall training accuracy is the binary accuracy across all the stations in the training set and is from the epoch where validation accuracy reached its peak. The validation accuracy is the accuracy achieved on the under-sampled evaluation set. The validation set accuracy is also the overall accuracy of the previously shown accuracy grids. The same applies to test sets.

For many rotations, there is a clear difference between the training, validation, and test performance. The comparative visibility estimator performs better on training data, or stations it has learned from. When stations are reserved as the validation or test set, their accuracies tend to be lower than the overall training accuracy. Batavia and Glen Falls are the exception, where the model has better performance on the Batavia validation and test sets than over the stations in the training sets. The model has better performance on the Glen Falls validation set than on the training set, however, this is not true for the Glen Falls test set.

Rotation	Overall Train Acc.	Valid Station	Valid Acc.	Test Station	Test Acc.
0	76.4%	Potsdam (POTS)	79.0%	Queens (QUEE)	75.8%
1	76.1%	Queens (QUEE)	78.8%	Batavia (BATA)	78.7%
2	73.5%	Batavia (BATA)	83.7%	Buffalo (BUFF)	71.5%
3	78.9%	Buffalo (BUFF)	68.9%	Elmira (ELMI)	69.7%
4	78.0%	Elmira (ELMI)	66.2%	Gabriels (GABR)	68.5%
5	75.4%	Gabriels (GABR)	74.6%	Glen Falls (GFAL)	71.7%
6	76.5%	Glen Falls (GFAL)	80.6%	Johnstown (JOHN)	70.2%
7	76.6%	Johnstown (JOHN)	70.8%	Manhattan (MANH)	72.5%
8	75.8%	Manhattan (MANH)	74.2%	Penn Yan (PENN)	73.9%
9	77.9%	Penn Yan (PENN)	72.1%	Potsdam (POTS)	76.1%

Table 5.1: Validation and Test set accuracy for each rotation.

Some stations show an opposite trend, where their validation and test set accuracies are significantly lower than the overall training accuracy. When Buffalo and Elmira comprise the validation and test sets, their accuracies do not even reach 70%. This reflects the results seen in the training sets. Some stations like Buffalo and Elmira have physical conditions that detrimentally affect performance. The model's ability to learn and generalize to these stations are largely impacted by the stations' short field of view, limited view of the sky, and high distances between the camera and visibility sensor.

While the under-performance of some stations may suggest over-fitting to the training set, the over-performance of some stations provide some evidence that some stations have a better baseline performance than other stations. Table 5.2 shows the difference in model accuracy on each station for the under-sampled training, validation, test sets.

The binary accuracy of examples where the query and reference images have the same visibility is not representative of the model's performance. As mentioned in the previous section, the accuracy in these cases is expected to be approximately 50%. In

Station	Training 2019 Acc.	Validation 2021 Acc.	Test 2022 Acc.
Batavia (BATA)	88.9%	85.9%	80.1%
Buffalo (BUFF)	77.0%	70.9%	72.8%
Elmira (ELMI)	73.9%	67.6%	71.3%
Gabriels (GABR)	84.9%	76.6%	70.1%
Glen Falls (GFAL)	85.2%	82.4%	73.7%
Johnstown (JOHN)	80.0%	73.2%	71.3%
Manhattan (MANH)	85.2%	76.5%	74.5%
Penn Yan (PENN)	83.7%	73.1%	75.4%
Potsdam (POTS)	82.7%	80.8%	78.0%
Queens (QUEE)	84.9%	80.6%	77.4%

Table 5.2: Validation and Test set equivalence exclusion accuracy.

order to get a deeper understanding the performance of the model on the different stations, I exclude the equivalence cases when calculating the accuracies for Table 5.2. I also use the under-sampled 2019 set for calculating the training accuracy for each station.

The table demonstrates a clear degradation of performance from when a station is used in the training set to when a station is used in the validation or test set. In seven out of the ten rotations, the model performs better when a station is in the validation set as compared to the test set. On average, there is a 5.9% difference in the model's performance between when a station is in the training set versus when it is in the validation set. Even more so, there is an average 8.1% difference in the model's performance between when a station is in the training set versus when it is in the test set. One reason could be that after training the model's weights are restored to the weights it had during the epoch with the highest validation accuracy. Since the validation accuracy does not increase much through the epochs, the saved model could be over-fitted with respect to the training and validation data or under-fitted with respect to the test data.

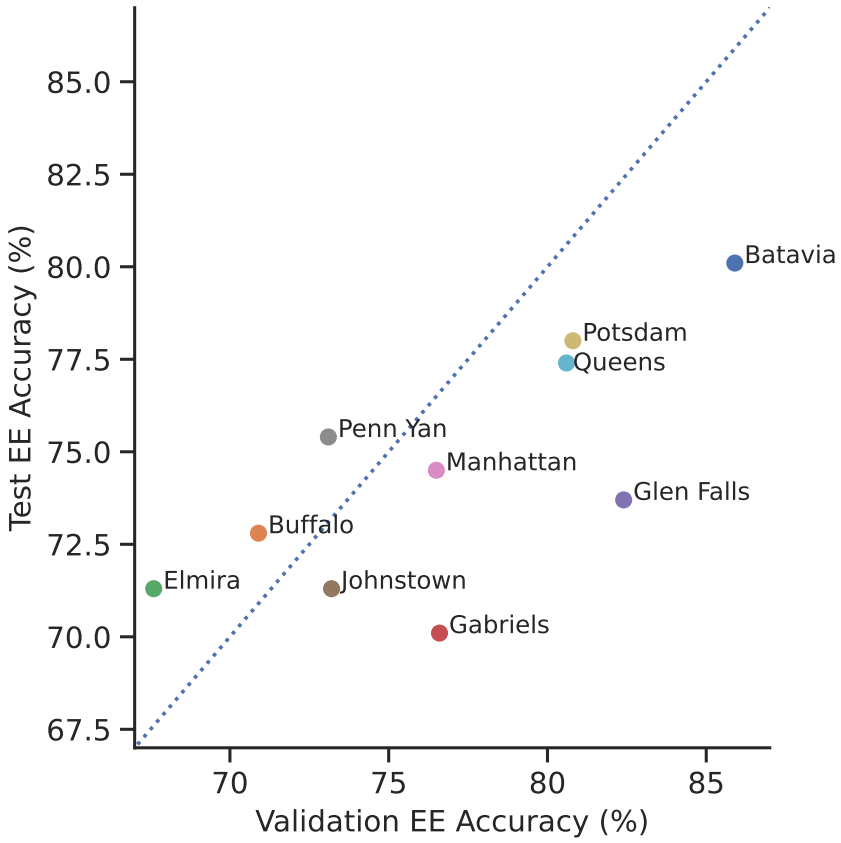


Figure 5.6: Test vs. Validation equivalence exclusion (EE) accuracy for each station.

Although there is a difference in performance between the validation and test sets, generally if a station has a higher accuracy in the validation set, the station will have a higher accuracy in the test set. This can be seen in Figure 5.6 where there is a slight linear relationship between validation and test set performance.

The difference in performance between the stations and the rotations could partially be explained by the location difference between the NYS Mesonet camera and the ASOS visibility sensor. In order to easily visualize how the location of the camera and visibility sensor affects performance, I computed a combined metric between the horizontal distance and elevation difference. I first normalize the horizontal distances and elevation difference for each of the camera, visibility sensor pairs between 0 and 1:

$$\text{normalize}(d_k) = \frac{d_k - \min(d_0, d_1, ..., d_k, ..., d_n)}{\max(d_0, d_1, ...d_k, ..., d_n) - \min(d_0, d_1, ...d_k, ..., d_n)} \ , \tag{5.1}$$

where I use the minimum and maximum distances or elevation distances. After normalizing, I average the normalized distance and elevation difference together. This produces a normalized metric that signifies how "far" the sensors are from each other for each station, where 0 signifies the camera and visibility sensor for a site are closer together than any other camera and visibility sensor among other stations, and 1 signifying that the camera and visibility sensor for a site are the furthest apart amongst other stations. Figure 5.7 shows the normalized combined distance metric plotted with respect to the test set accuracy for each station. Stations with overall high performance in the test set tend to have camera and visibility sensors that are closer together, while stations with an overall low performance as the validation and test sets tend to have camera and visibility sensors that are closer further apart. This due to the de-correlation between the visibility measured by ASOS sensor and the visual phenomena present in the NYS Mesonet camera image. As these sensors are further apart, the de-correlation grows.

Queens is an exception to this pattern, where although the camera and visibility sensor are fairly far away the model is still able to achieve a fairly high accuracy. This is because of two reasons. The first reason is that the Queens station has a long and wide field of view with visible landmarks up to 9 miles away. The second reason is that Queens is located in a coastal area of New York. The coastal fog may be more uniform, wide spread, and lingering than advection or localized fog found in other environments. Because of the persistent quality of coastal fog, there may be more correlation between

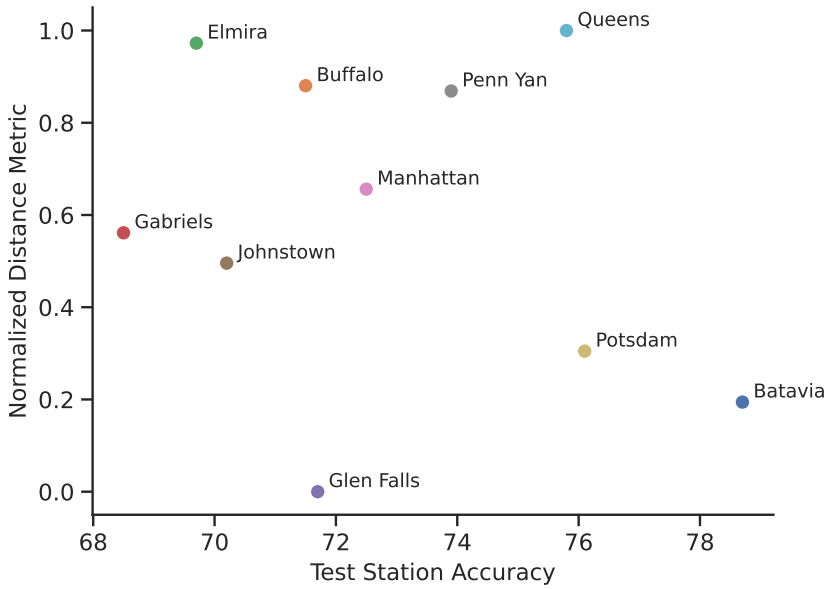
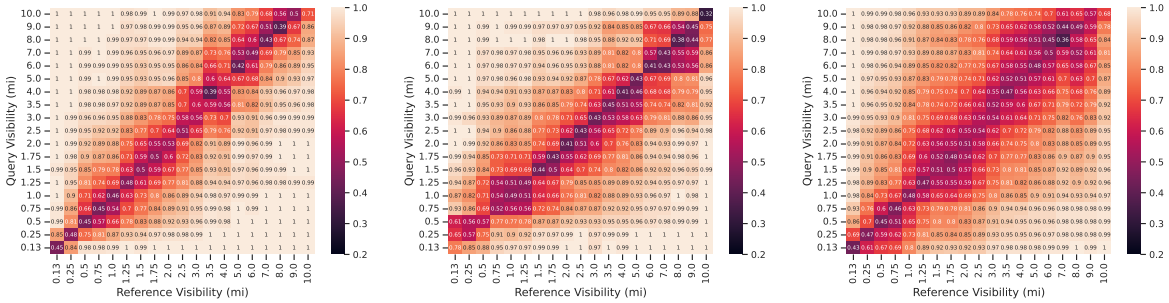


Figure 5.7: Normalized distance metric vs. test set accuracy.

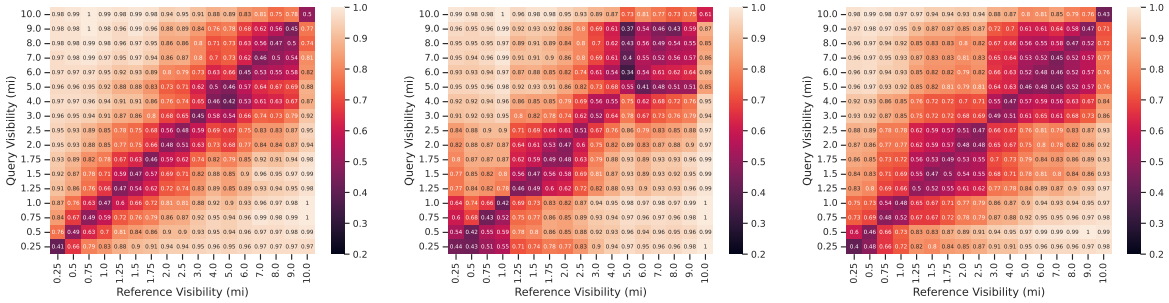
the camera and visibility sensor even with their high distance and elevation difference. Glen Falls is another exception where it has low test accuracy and have a camera and ASOS sensor that are close together. Glen Falls has a high accuracy in the validation set, in addition, has a very short field of view which could be impacting its performance when in the test set.

Stations Batavia, Potsdam, Queens, and Glen Falls are the best examples of cross-station generalization. As Batavia changes from being in the training set to the validation set, there is some degradation in performance, as shown in Figure 5.8, yet, the model is still able to distinguish between difference visibility distances with reasonable accuracy. However, there is additional degradation in performance when Batavia is in the test set, where different visibility distance clusters become more difficult to distinguish between, such as 1 to 2.5 miles and 3 to 10 miles. Potsdam also shows a similar pattern were visibility clusters of weak performance begin to form as a station moves



(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure 5.8: Batavia accuracy grids across training, validation, and test years.

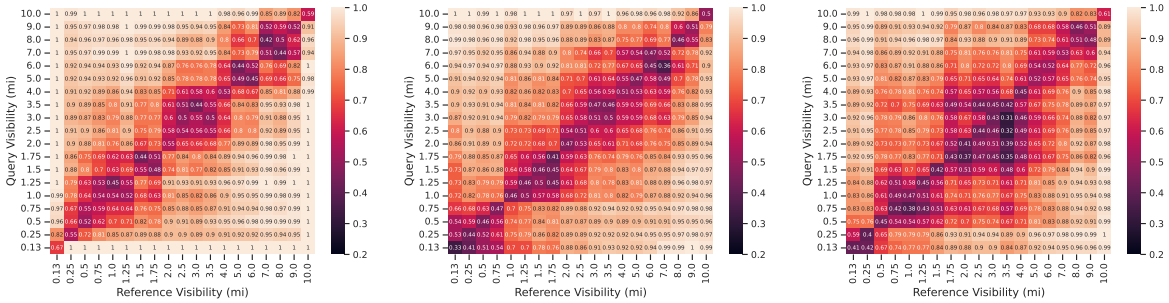


(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure 5.9: Potsdam accuracy grids across training, validation, and test years.

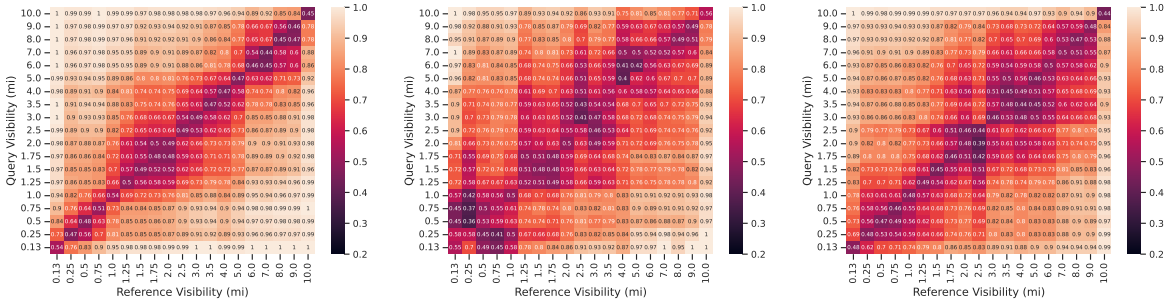
from the training set to a validation or test set. In these low performance visibility clusters the model is less likely to distinguish between finer visibility gradations (Figure 5.9. This is strongly apparent in when Potsdam is in the test set. There are two prominent square regions of decreased performance, the first in the range of 1.25 to 2.5 miles and the second in the range of 3.0 to 9.0.

Queens shows similar behavior not just in the validation and test sets, but in the training set as well (see Figure 5.10). Similarly, Penn Yan shows these clusters of low performance as well as specific visibility values that tend to be difficult to predict on (see Figure 5.11). These bands of low performance tend to occur between 1 and 2 miles, where the step difference between visibility readings is quite small. These bands also



(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure 5.10: Queens accuracy grids across training, validation, and test years.

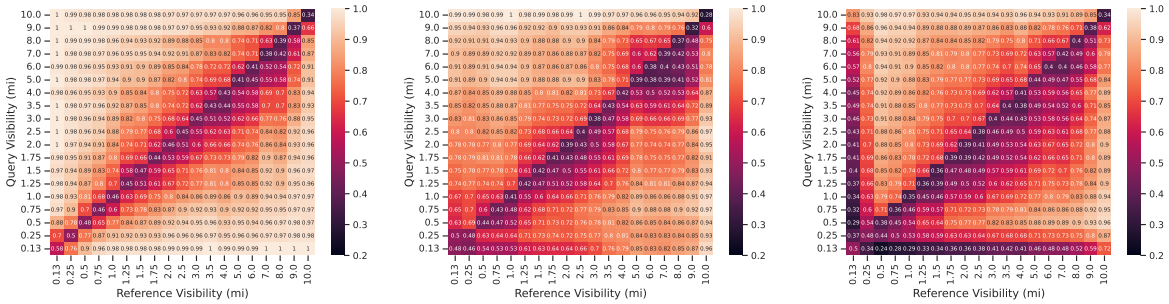


(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure 5.11: Penn Yan accuracy grids across training, validation, and test years.

tend to occur between 5 and 9 miles, where the step difference between the visibility readings might be large, however, the visual difference in the image could be small or non-existent.

Some stations that had low accuracies when in the training set, such as Buffalo, Elmira, and Johnstown, suffered a further degradation of performance when in the validation and test set. However, some stations demonstrated unique results in their performance. When Gabriels is in the test set, results for query and reference images with a 0.13 visibility distance produced accuracies so low the model was performing worse than random chance (Figure 5.12c).



(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure 5.12: Gabriels accuracy grids across training, validation, and test years.

Upon further investigation of images with a 0.13 visibility distance in 2022, there seems to be an unusually high amount of mislabeled examples. Figure 5.13 shows various images of drastically different visibility conditions that were all labeled as 0.13 miles by the nearest ASOS visibility sensor. At Gabriels, the closest mountain peak in the background is approximately 2 miles away while the furthest mountain peak is approximately 10 miles away. In addition, the closest tree line is approximately 300 feet away from the camera, and the slightly further tree line is approximately 0.5 miles from the camera. The image closest to a true visibility distance of 0.13 is Figure 5.13a, while the rest have incorrect labels. The model's inability to predict on pairs including images with a visibility distance of 0.13 miles may have been a combination of inability to generalize and incorrect labels found in 2022.

This issue is not exclusive to Gabriels. Many stations have mis-labeled examples through out different visibility bins, especially the low visibility cases. This further contributes to the model's low performance near the diagonal for many stations. However, this highlights one of the strengths of the comparative visibility approach. Since, the data set has a significant amount of noise, the true labels do not need to be exact in order for the model to learn comparative visibility. If a query image is labeled with a visibility distance of 0.13, however has a true visibility distance of 3 miles, and a

(a) (b)

(c) (d)

Figure 5.13: Images from the Gabriels NYS Mesonet station during 2022 where the closest ASOS station reported a visibility of 0.13 miles.

reference image is labeled with a visibility distance of 10 miles, however, has a true visibility distance of 6 miles, both images have a significant amount of noise in the label, however, the model's prediction should be the same. The comparative approach allows the model to learn how to distinguish visibility in context of highly noisey data.

5.2 Numerical Visibility Prediction

5.2.1 Temporal Generalization Results

During cross-validation, the years 2021 and 2022 were reserved exclusively for validation and test sets in order to assure temporal independence between the folds. Since entire years of data were reserved, the validation and test years of stations in the training set

can be evaluated to quantify temporal generalization of the models. This demonstrates how well the model works on previously seen stations, however, unseen years.

Figure 5.14 shows the spread of numerical visibility predictions of Batavia for each true visibility distance bin. When Batavia is in the validation set (examples belonging to the year 2021), the model generally predicts a numerical visibility distance within range of the true visibility distance. However, as the true visibility distance increases, the error also increases (full breakdown of RMSE for each station and visibility value is in Appendix B.2).

One of the reasons for the high variance in visibility distance prediction for each of the unique visibility bins is the low correlation between the visibility phenomena captured by the camera image and the measurement from the visibility sensor. In Figure 5.15, all query images along the left hand column have a true visibility distance of 1.5 miles. The right column shows the output probabilities, in blue, of the comparative visibility estimator when the image is compared to the set of reference images. The orange line is the fitted sigmoid, and the red star is the numerical visibility prediction for the query image.

Panel (a) is the image with the smallest numerical prediction amongst query images with a reported visibility distance of 1.5 miles. Panel (c) has a numerical visibility prediction that matches the 25th percentile of all visibility predictions for images with a 1.5 miles visibility distance. Panel (e) has the median prediction of all visibility predictions for images with a 1.5 miles visibility distance. Panel (g) has a numerical visibility prediction that matches the 75th percentile of all visibility predictions for images with a 1.5 miles visibility distance. Finally, panel (i) is the image with the largest numerical prediction amongst query images with a reported visibility distance of 1.5 miles.

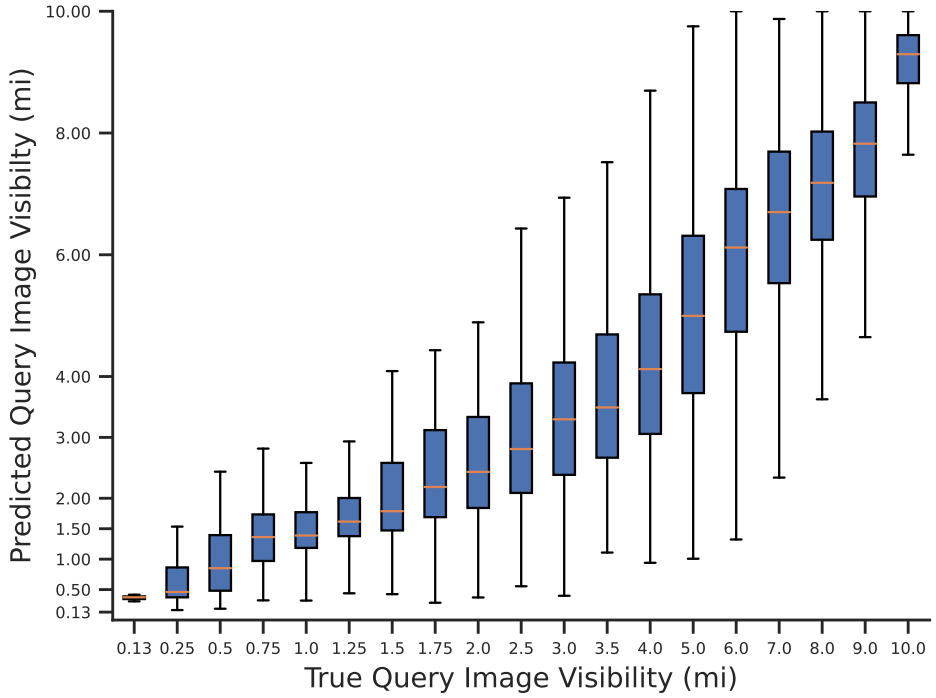
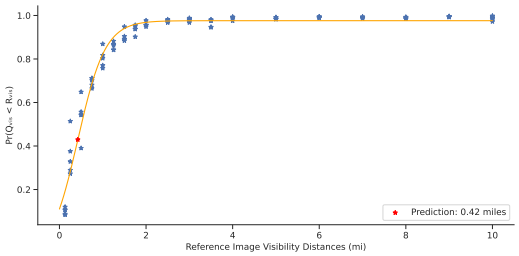


Figure 5.14: Boxplot for numerical predictions on Batavia 2021 using model from rotation 0.

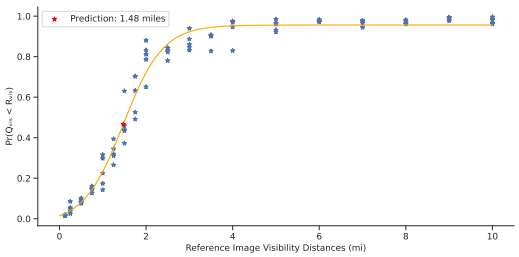
In the image with the smallest numerical prediction (a), the slightly visible tree line is less than a half mile from the camera. The model's prediction of 0.42 is reasonable, even though the reported ASOS visibility measurement is over 3 times that value. The image with the median numerical prediction (e) is a good example where the ASOS visibility reading and the visual features in the image correlate. The tree line that is less than a half mile away from the camera is visible, although foggy, signifying some reduction in visibility, however, greater than half a mile. The model predicts 1.78 miles, which is reasonably close to 1.5 miles for practical purposes. The image with the numerical prediction closest to the 75th percentile (g) shows a very similar seen to the previous image (e), but has a much higher visibility prediction. One possible reason is the lack of snow in image (g). Most low visibility examples correlate with

(a) Minimum



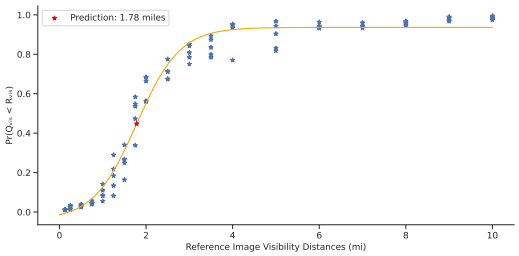
(b)

(c) 25th Percentile



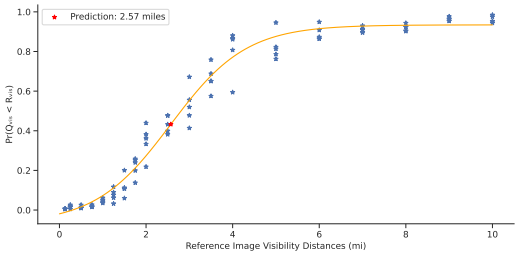
(d)

(e) Median



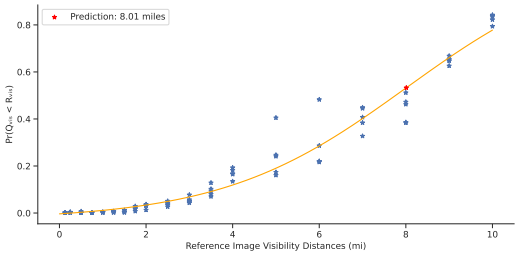
(f)

(g) 75th Percentile



(h)

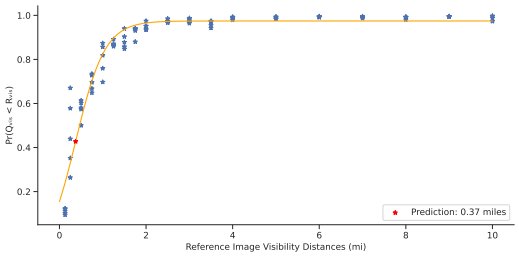
(i) Maximum



(j)

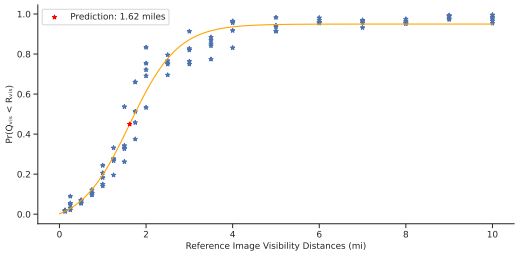
Figure 5.15: Predictions for Batavia 2021 and query images with a reported visibility distance 1.5 miles.

(a) Median



(b)

(c) Maximum



(d)

Figure 5.16: Predictions for Batavia 2021 and query images with a reported visibility distance of 0.13 miles. While all images have the same reported visibility, there are clearly different visibliity conditions depicted in each of the images.

some sort of weather, albeit precipitation or snow. These weather conditions are more likely to occur in the winter months where there tends to be less greenery and more seasonal conditions like snow cover. In the image with the largest numerical prediction (i) the tree line is fairly clear with some localized mistiness, however, the mistiness seems to be due to some moisture on the camera lens. The model is robust to this camera occlusion and still accurately predicts a high visibility distance of 8 miles, even though the ASOS station is reporting 1.5 miles.

Moreover, there often is a de-correlation between the camera images and the visibility measurements. This is especially noticeable among lower visibility conditions. Figure 5.16 depicts the median and maximum prediction for query images with a reported visibility of 0.13 miles. The image with the median prediction (a) has a scene is truly saturated with fog, thus has an extremely reduced visibility of less than half a

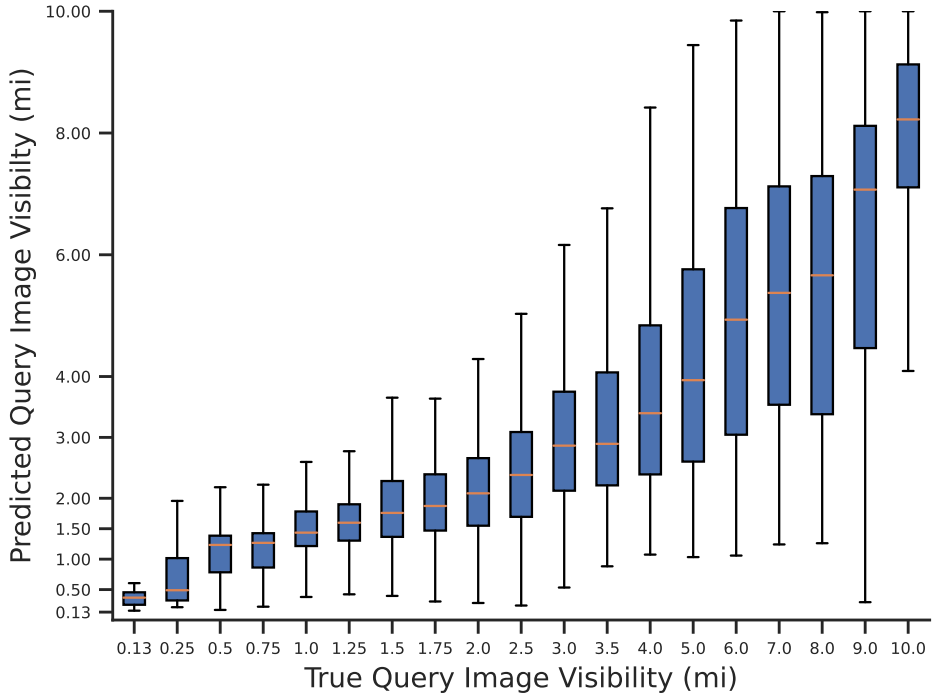
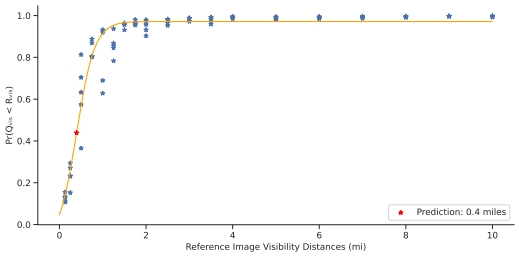


Figure 5.17: Boxplot for numerical predictions on Batavia 2022 using model from rotation 0.

mile, while, the image with the maximum prediction (c) is more clear where the half mile tree line is visible although slightly foggy. The model's predictions reflect what is visually present in the images, where image (a) has a predicted visibility distance of 0.37 miles and image (c) has a predicted visibility distance of 1.62 miles. These predictions reflect visually what is captured by the Mesonet camera than the ASOS reported visibility of 0.13 miles.

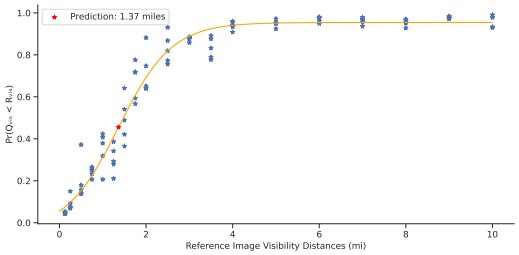
These results are also reflected in the test set. The spread of numerical visibility predictions is similar in the test set than in the validation set (see Figure 5.17). As shown in Figure 5.18, all images along the left hand column are examples where the reported visibility distance is 1.5 miles. All examples except the 25th percentile image (c) have reasonable visibility estimates that are more representative of what is depicted

(a) Minimum



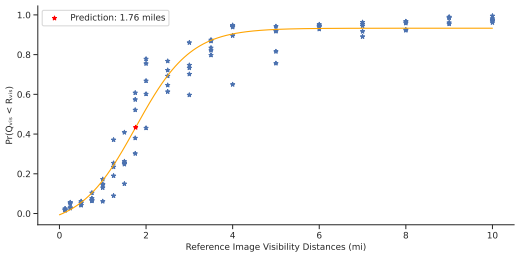
(b)

(c) 25th Percentile



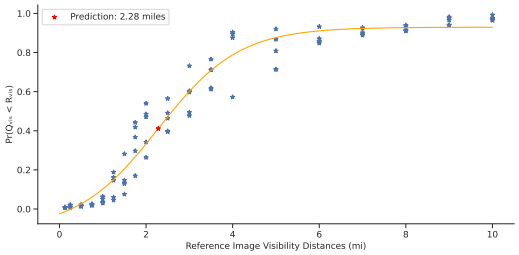
(d)

(e) Median



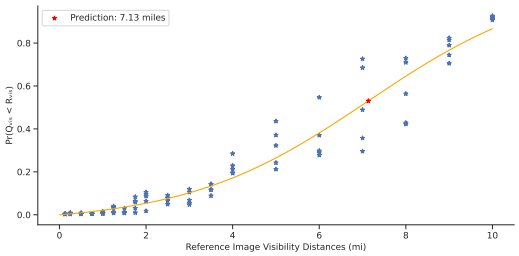
(f)

(g) 75th Percentile



(h)

(i) Maximum



(j)

Figure 5.18: Predictions for Batavia 2022 and query images with a reported visibility distance 1.5 miles.

in the image than the reported ASOS visibility distance. The 25th percentile example (c) has an obscured lens with impacted snow. Since there is no automatic filtering of obscured images, they exist within the training set. The model predicts a numerical visibility estimate of 1.37 miles, which is not depicted by the image. However, the model's output probabilities are not spread widely. Since obscured cases exist within the training set, the model may learn that snowy conditions that lead to an obstructed lens correlates to a reduced visibility distance.

The numerical visibility prediction results for Gabriels tells a similar story. Like Batavia, the range of predictions increase as query image visibility increases. In addition, for visibility distances like 4.0 miles, predictions can range from 0.13 miles to 7.0 miles (Figure 5.19). Like Batavia, some of this is due to the desparity between the camera image and the visibility measurements. Figure 5.20 shows the minimum, 25th percentile, median, 75th percentile, and maximum predictions for query images with a reported visibility of 2.0 miles. Image (a) depicts a scene where the secondary tree line that is approximately 0.3-0.5 miles from the camera is completely obscured by fog. The model predicts 0.17 miles for the image's visibility distance, albeit the reported visibility is 2.0 miles. The 25th percentile and median predictions have images where the secondary tree line is visible. However, the closest mountain (which is approximately 5.0 miles away) is slightly visible, but perhaps not entirely distinguishable from the sky. The model's predictions of 1.37 and 1.83 miles, respectively, are reasonable for the query images. The 75% prediction has an image where the closest mountain is visible. Although the prediction of 2.44 miles is higher than the predictions for images (c) and (e), it is a miss-estimation of the true visibility, which should be closer to 5.0 mile. However, again, the true visibility in the camera image does not match the reported 2.0 miles. Finally, the maximum prediction is fairly accurate given the

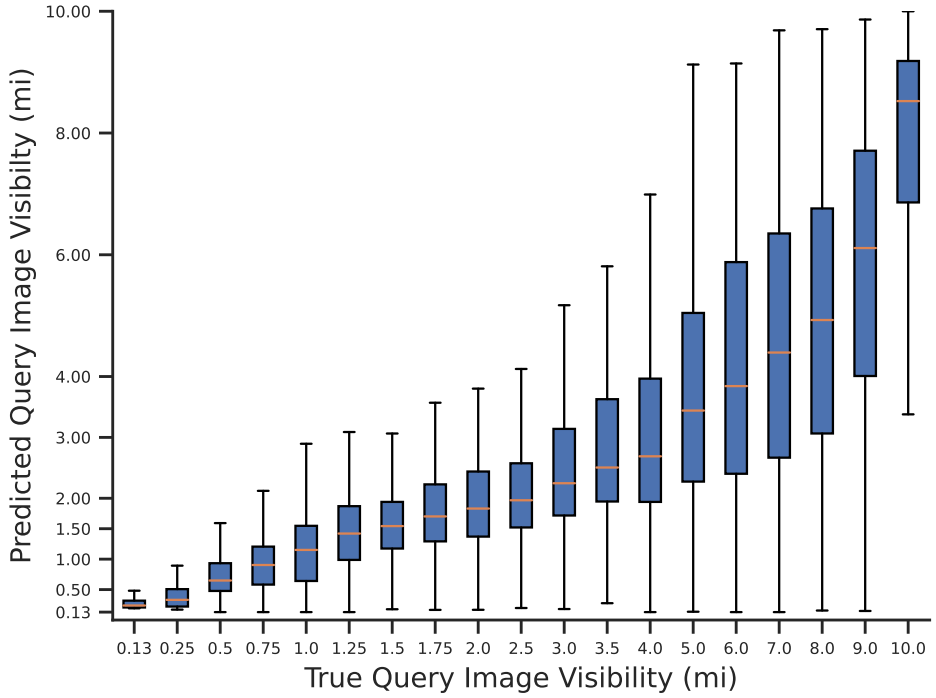
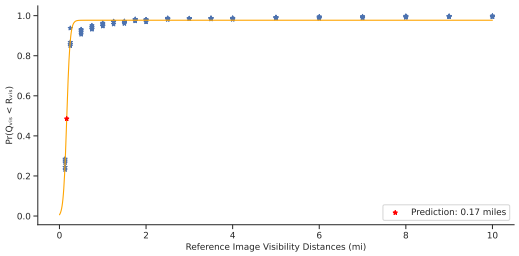


Figure 5.19: Boxplot for numerical predictions on Gabriels 2021 using model from rotation 3.

query image of (i). The mountain range, which is 8-9 miles away from the camera at its furthest point) is completely distinguishable from the sky and the model predicts an appropriate visibility of 8.43 miles.

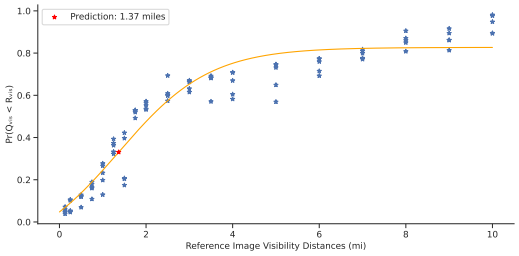
Not all images have a clear visibility estimate. Fog can be localized to specific regions, especially advection fog which hovers close to the ground. This is particularly visible in (a) and (c) of Figure 5.21. Overall, the images look clearer than the reported visibility of 0.25 miles. The model agrees and predicts low visibility values of 0.17 and 0.22 miles, respectively. I believe this is due to the high amounts of low lying, morning fog that is frequently present at the Gabriels station. This is particularly present in the training set. Figure 5.22 shows that 0.25 mile visibility often occurs at 10 or 11

(a) Minimum



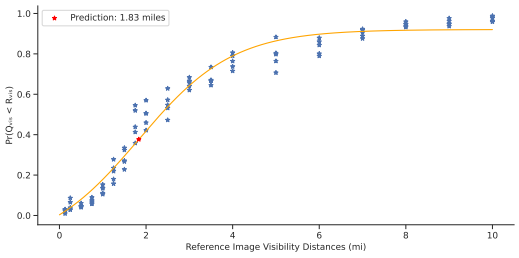
(b)

(c) 25th Percentile



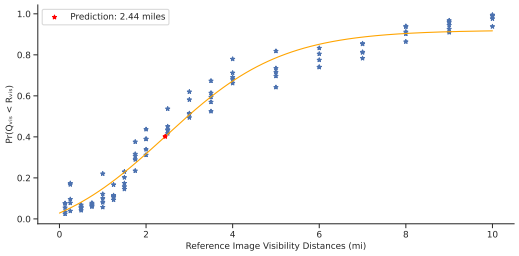
(d)

(e) Median



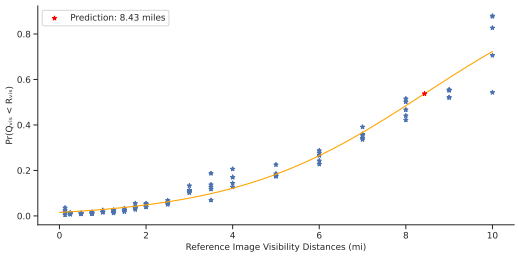
(f)

(g) 75th Percentile



(h)

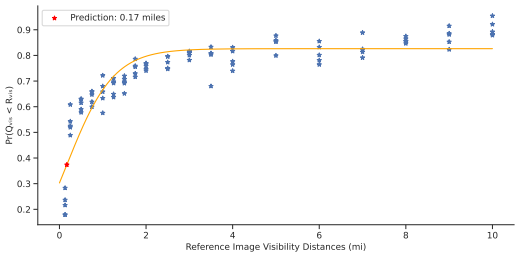
(i) Maximum



(j)

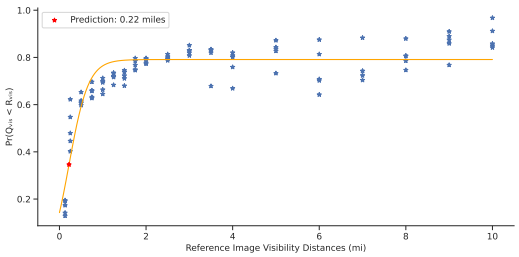
Figure 5.20: Predictions for Gabriels 2021 and query images with a reported visibility distance 2.0 miles.

(a)



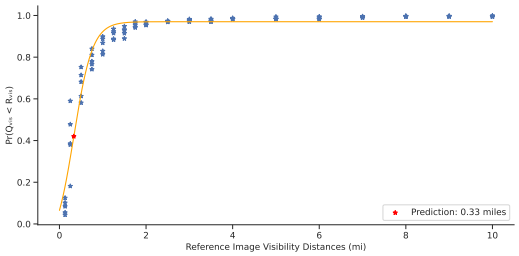
(b)

(c)



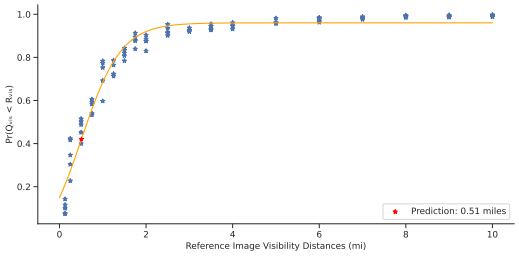
(d)

(e)



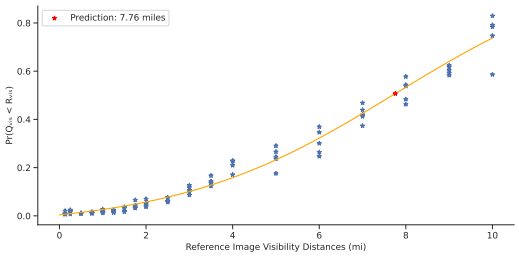
(f)

(g)



(h)

(i)



(j)

Figure 5.21: Predictions for Gabriels 2021 and query images with a reported visibility distance 0.25 miles.

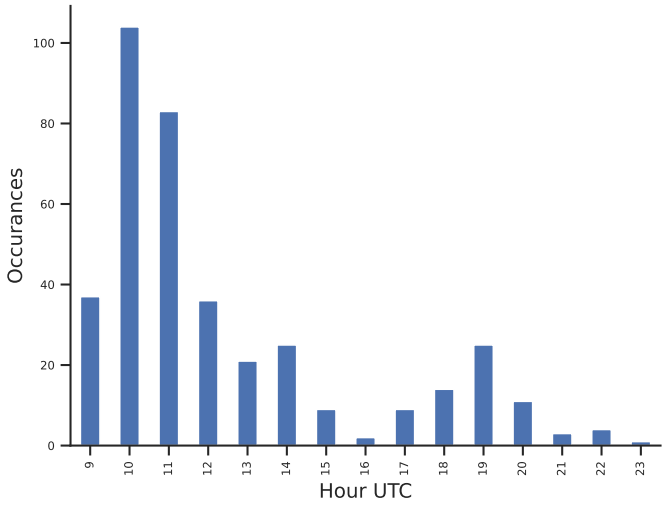


Figure 5.22: Histogram of which hours of day that 0.25 mile visibility is reported.

UTC (6am or 7am local time) during the year of 2019. The fog that occurs is most likely the field condensation that happens in the morning. The ASOS station is almost 30 meters lower in elevation than the NYS Mesonet station. Due to this difference in elevation, it is likely that the ASOS station is picking up on more dense, low-lying fog than what is present at the NYS Mesonet station. This affects how the model identifies low visibility conditions for Gabriels, and results in images with advection fog being flagged with lower visibility than what is actually depicted in the camera lens.

5.2.2 Cross-site Generalization Results

Although temporal generalization is important, the comparative visibility estimator was designed to be a general visibility estimator in order to assess the visibility of any location. The results in this section are for the validation and test sets for different rotations.

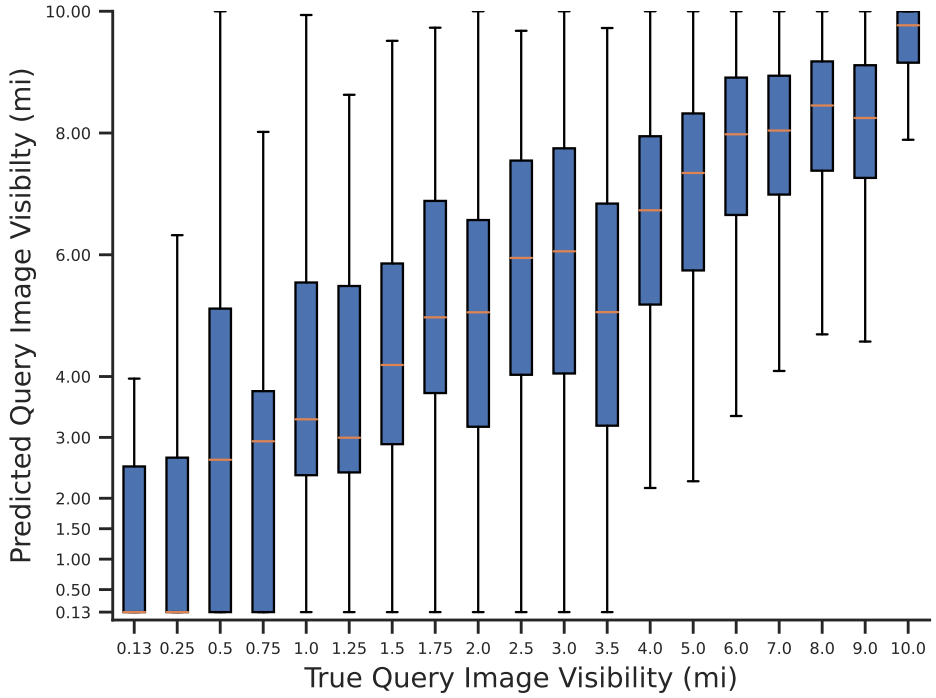
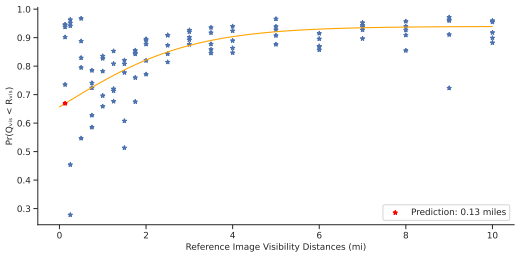


Figure 5.23: Boxplot for numerical predictions on Queens 2022.

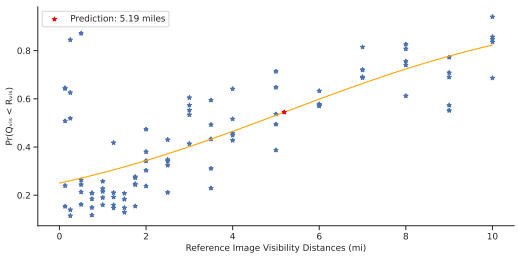
Like the results shown in the previous section, the numerical visibility predictions often vary widely. However, for most stations, as the query image's visibility increases, so does the prediction (see Appendix B.3 for full breakdown). This is true for Queens when it is not included in the training set, but rather is the test station (Figure 5.23). However, for the Queens station the variance of predictions is quite high for many of the visibility distance levels. Like before, it can be helpful to look at the minimum, 25th percentile, median, 75th percentile, and maximum predictions. In Figure 5.24, all of the query images have a reported visibility of 4.0 miles, however, visibility predictions range from smallest reportable visibility of 0.13 miles to the highest reportable visibility of 10.0 miles. Upon visual inspection of each of the images, the model's predictions are reasonable, if not fairly accurate in measuring the actual visibility of each image.

(a) Minimum



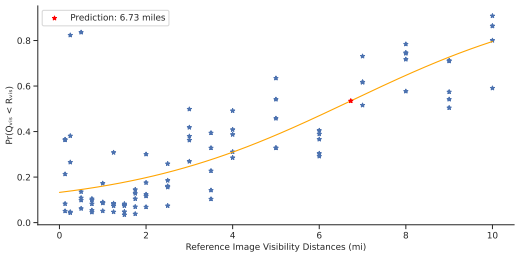
(b)

(c) 25% Percentile

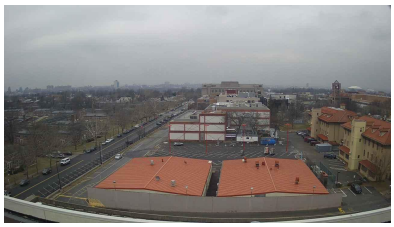


(d)

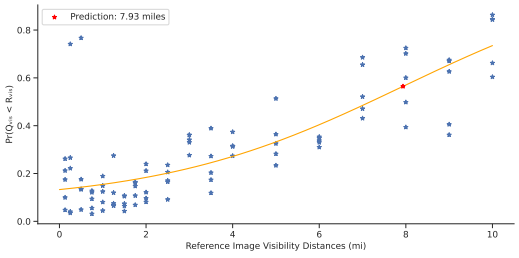
(e) Median



(f)

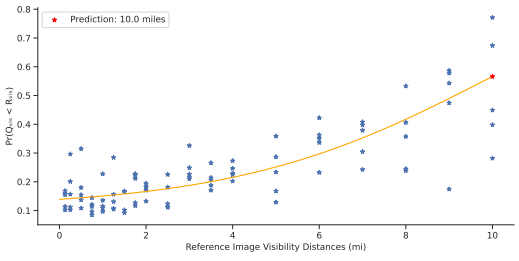


(g) 75% Percentile



(h)

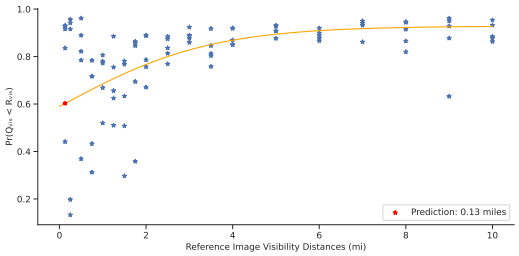
(i) Maximum



(j)

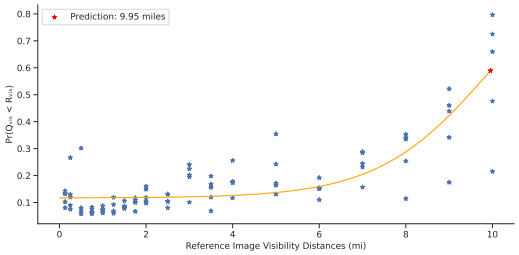
Figure 5.24: Predictions for Queens 2022 and query images with a reported visibility distance 4.0 miles.

(a) Median



(b)

(c) Maximum



(d)

Figure 5.25: Predictions for Queens 2022 and query images with a reported visibility distance 0.13 miles.

Furthermore, visibility level 0.13 miles has an unusually high variance for its numerical visibility predictions, especially when compared to 0.25 miles. However, when looking at the images associated with the median and maximum predictions, the model's assessment of the underlying visibility is more accurate than the reported ASOS visibility (Figure 5.25).

Not all stations had such a high variance of prediction values. Some stations had the opposite where the model seemed predict visibility values within a small range for all query images. When Elmira was designated as the test station, the model predicted visibility values with a median of approximately 2.0 miles for most visibility distance groups (Figure 5.26). Even more unusual, there was more variance in the prediction values among query images with lower visibility distances than higher visibility distances. However, if we look at the minimum, 25th percentile, median, 75th percentile,

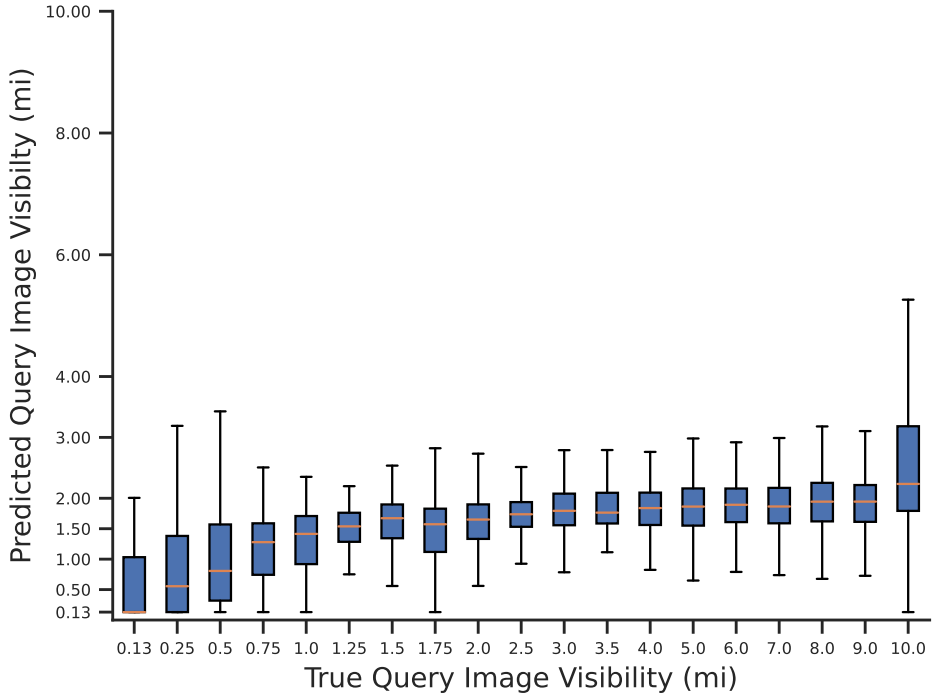
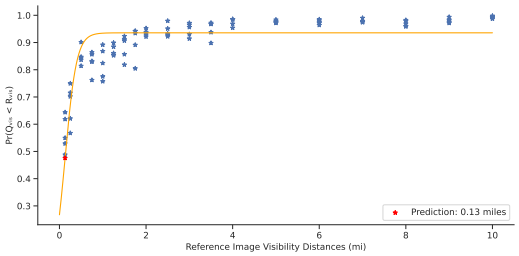


Figure 5.26: Boxplot for numerical predictions on Elmira 2022.

and maximum predictions for query images with a reported visibility of 3.0 miles, the model seems to be making reasonable predictions (Figure 5.27).

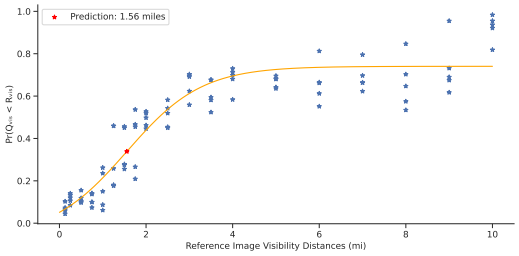
One extremely unfortunate aspect of Elmira is that most of the scene is occupied by a tree line that is extremely close to the camera. There are little to no other features that can be used to gauge visibility distances much farther than 0.5-1 mile. In the highly obscured example (a), the model accurate predicts the lowest reportable visibility value. In the images with the numerical prediction closest to the 25th percentile (c), median (e), and 75th percentile (g) the visibility appears to be relatively the same. The model predicts a visibility distance around 1.5 miles to 2.0 miles for each of these images. Although the reported visibility for those images is 3.0 miles, there are no major visual cues to distinguish 1.5 miles of visibility to 3.0 miles of visibility. The model predicted a maximum of 6.36 miles (i) which depicts a bright day with a strikingly blue sky. The

(a) Minimum



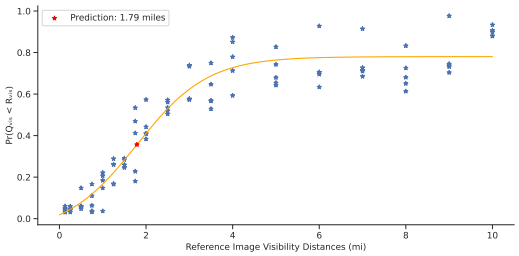
(b)

(c) 25th Percentile



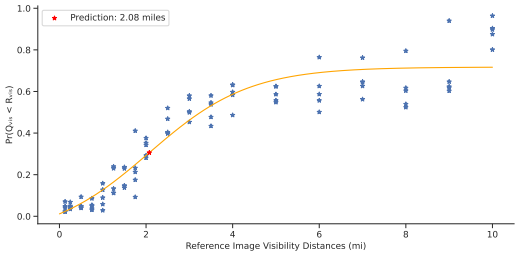
(d)

(e) Median



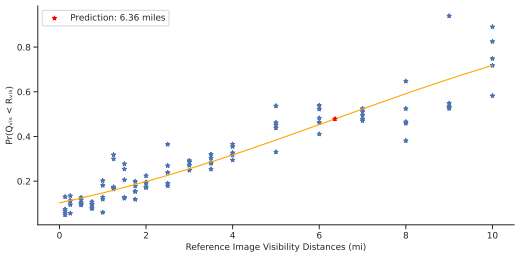
(f)

(g) 75th Percentile



(h)

(i) Maximum



(j)

Figure 5.27: Predictions for Elmira 2022 and query images with a reported visibility distance 3.0 miles.

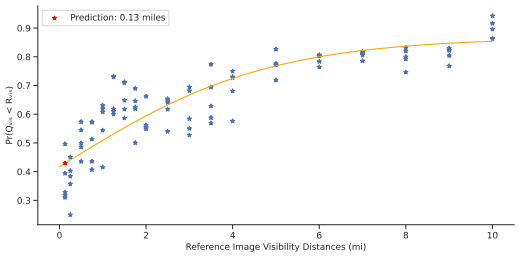
image with the maximum numerical prediction (i) probably has a true visibility of 10 miles, however, there is no way to visually distinguish between 6 miles and 10 miles for this particular site.

When looking at the 25th percentile, median, 75th percentile, and maximum predictions for query images with a reported visibility of 10.0 miles, it becomes clear that the maximum discernible visibility for Elmira is somewhere close to 3.0 miles (Figure 5.28). Image (a) and (i) show the most extreme predictions for query images of 10 miles. Although the prediction for image (a) is wrong and the prediction for image (i) reaches up to 8 miles, this is representative for the majority of predictions for query images of 10 miles. When looking at the 25th percentile, median, and 75th percentile predictions, it demonstrates that when Elmira is experiencing high visibility conditions, the model predicts around the 2.0 to 3.0 mile range.

This is further supported with looking at the reference images for 2.0, 3.0, and 4.0 miles (Figure 5.29). The reference image with a reported visibility distance of 2.0 miles has a slight hazy to it, but in reference images for 3.0 and 4.0 miles the tree line is completely distinguishable from the sky and there is no apparent fog present in the scene. Due to these limitations, the model is not able to reliably predict higher than 3 miles of visibility for the Elmira station.

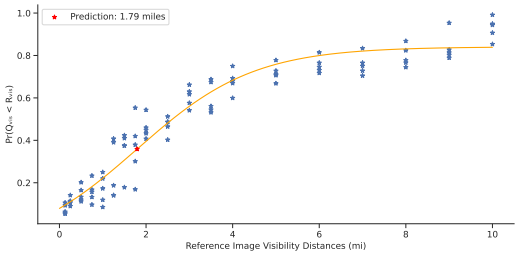
Some stations prove to be especially difficult to generalize. The model's ability to generalize to the Manhattan station was significantly different depending whether it was in the validation set or test set (Figure 5.30). Manhattan is a unique station where one of the main issues is that, in low visibility conditions, fog is not always uniformly distributed across the scene. For example, the image with the maximum numerical prediction (i) in Figure 5.31 shows a fairly clear day where all of the buildings in the

(a) Minimum



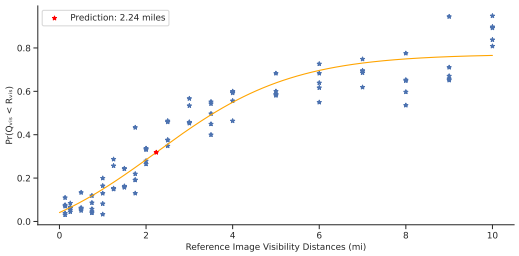
(b)

(c) 25th Percentile



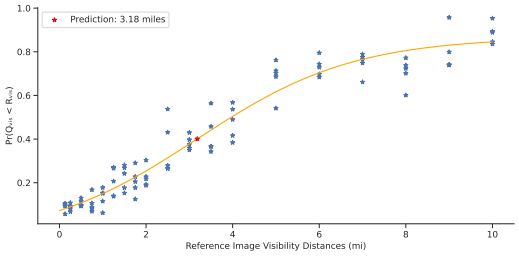
(d)

(e) Median



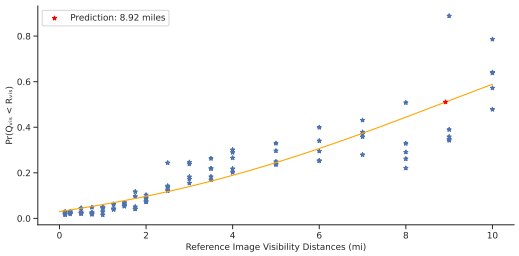
(f)

(g) 75th Percentile



(h)

(i) Maximum

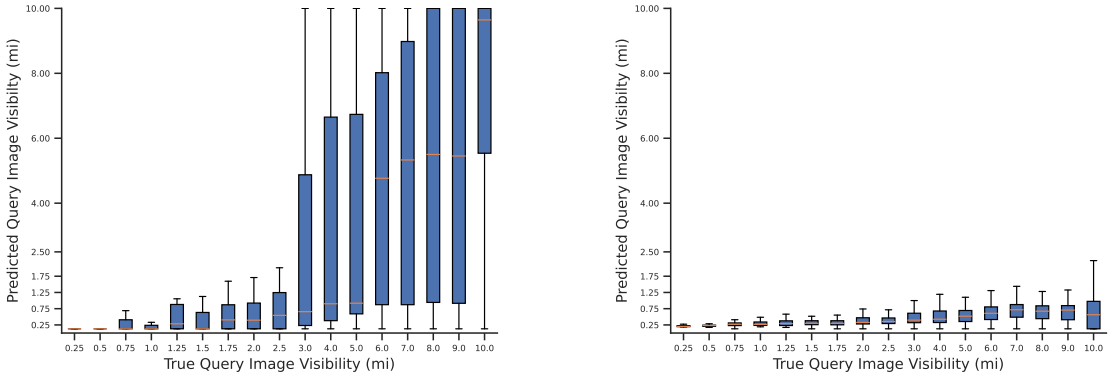


(j)

Figure 5.28: Predictions for Elmira 2022 and query images with a reported visibility distance 10.0 miles.

(a) 2.0 miles (b) 3.0 miles (c) 4.0 miles

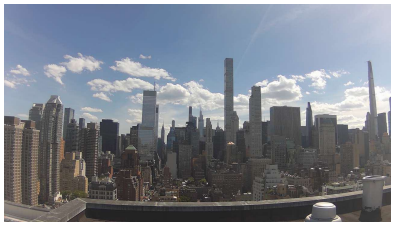
Figure 5.29: Selection of references images for Elmira with visibility distances (a) 2.0 miles (b) 3.0 miles and (c) 4.0 miles.



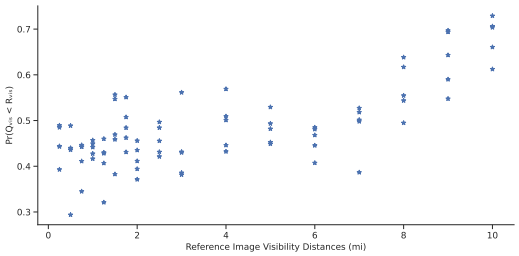
(a) (b)

Figure 5.30: Boxplot for numerical predictions on Manhattan (a) 2021 and (b) 2022.

scene are present. However, in image with the median prediction (e) the top of long skinny building (432 Park Ave) is obscured. This often happens in reduced visibility cases at the Manhattan station. The tops of the skyscrapers are obscured when fog rolls in, so the visibility distance is different in the top half of the scene than the bottom half. Images with the 25th percentile (c) and maximum (i) numerical prediction show where the model is accurately assessing the depicted visibility. However, the images with the median (e) and 75th percentile (g) numerical predictions have some fog, however, the buildings in the distance that are at least 2.0 miles away are still visible. The model under-predicts in these cases, perhaps due to the non-uniform fog. Sometimes the sigmoid does not fit well to the comparative visibility estimator probabilities and

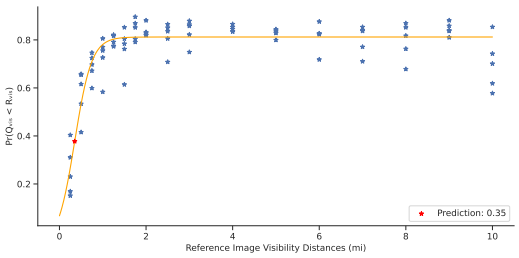


(a) Minimum



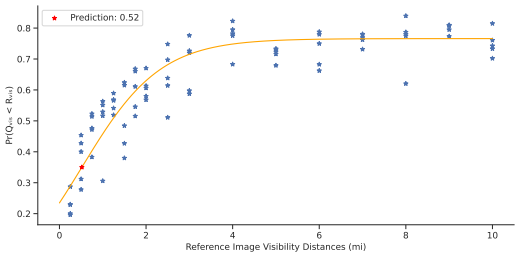
(b)

(c) 25th Percentile



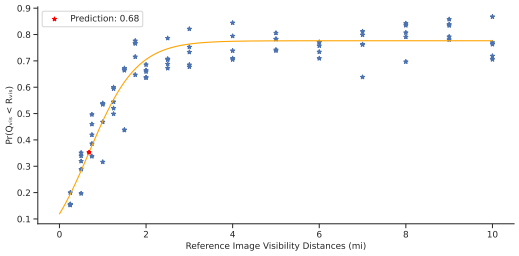
(d)

(e) Median

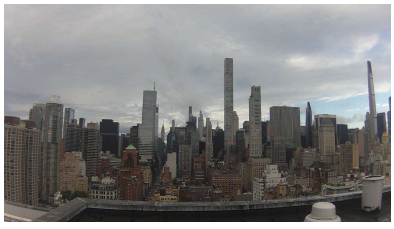


(f)

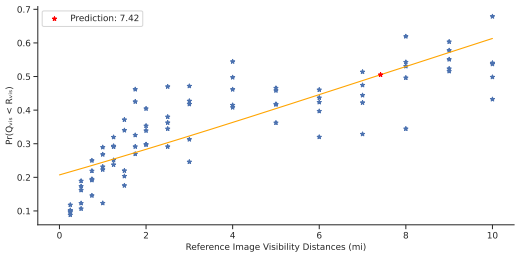
(g) 75th Percentile



(h)



(i) Maximum



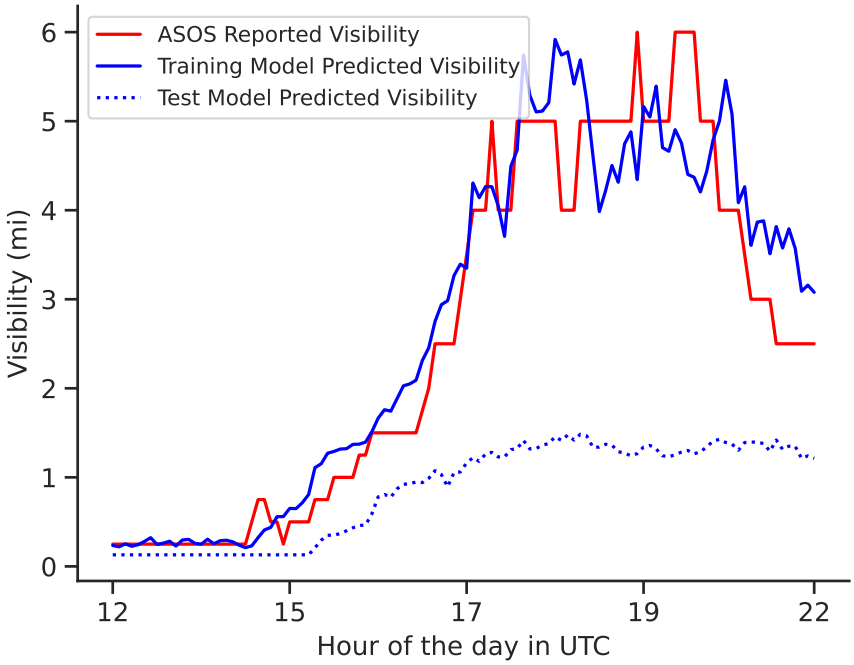
(j)

Figure 5.31: Predictions for Manhattan 2022 and query images with a reported visibility distance 5.0 miles.

the model is unable to predict a numerical visibility value. In this case, the model is struggling to generalize to the Manhattan station, so there is often more variance in the comparative visibility estimator probabilities.

Ultimately, this approach is designed to be used as a visibility monitoring tool, so its important to look at the performance of the models over different time periods. Figure 5.32a shows the reported visibility readings for the Batavia station for January 13th, 2022. The red line shows the ground truth ASOS reported visibility. The blue line shows the visibility predictions from a model trained with Batavia in the training set, and the blue dotted line shows the visibility predictions from the model trained where Batavia was in the test set. The images underneath the plot show sample images from different hours of that day. The model where Batavia was in the training set follows the ASOS measurements closely over 10 hours of the day. The model which was trained while Batavia was in the test set captures the general trend, but struggles with estimating the high visibility cases. However when examining the sample images from the different hours of the day, the model from the rotation where Batavia was in the test set does quite well considering the visual features present in each of the sample images. The tree line in the image is approximately a half mile away from the camera, so images from hours 17 to 22 all look pretty similar. The model with Batavia in the training set may be able to pick out the subtle variations in the sky to determine higher visibility cases, however the model with Batavia in the test set cannot.

Figure 5.33a shows the visibility readings and model predictions for Queens on November 4th, 2022. Here the model predictions both from the model where Queens was in the training set and the model where Queens was in the test set closely follow the ASOS readings. The Queens station has a long field of view, with buildings in frame



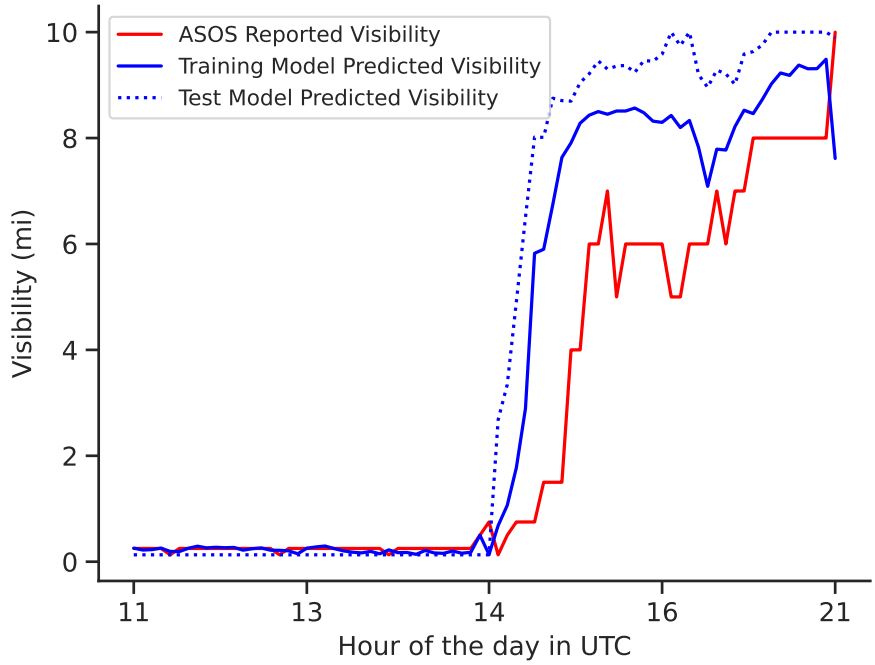
(a)

(b) Hour 12 (c) Hour 15 (d) Hour 17 (e) Hour 19 (f) Hour 22

Figure 5.32: Visibility readings and predictions for (a) Batavia on January 13th and sample images from different hours of the day (b)-(f).

that are up to 9 miles away. Unlike Batavia, there are many features and landmarks in the Queens station camera view that can signify visibility distances of larger values.

However, not every station demonstrates an ability to follow the ASOS measurements throughout the day. Figure 5.34a shows the visibility readings and model predictions for Elmira on November 11th, 2022. As the visibility varies between 2 and 10 miles throughout the day, both the training and test model's prediction are stable around 2 miles. This is primarily due to the very short field of view at Elmira, where

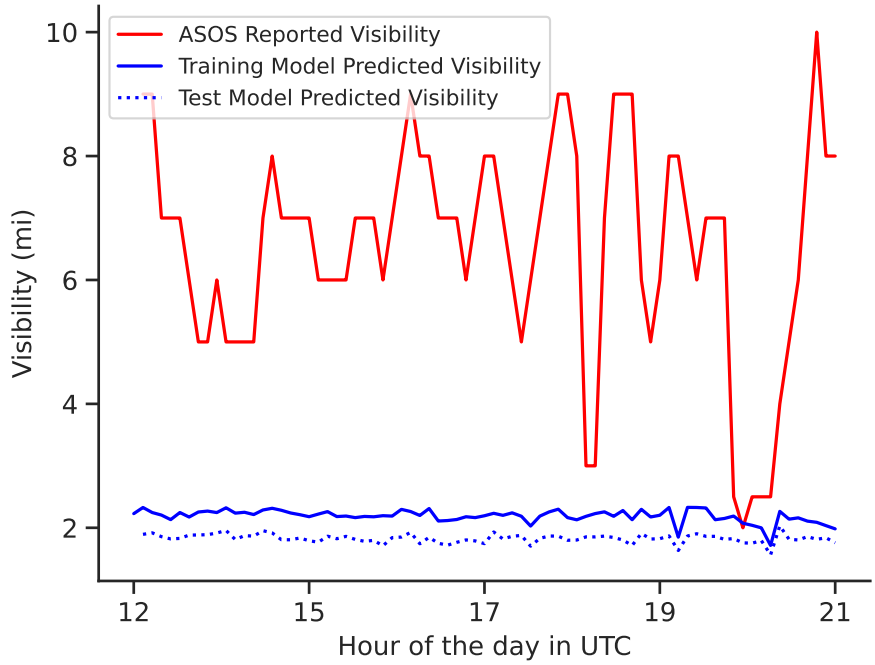


(a)

(b) Hour 11 (c) Hour 14 (d) Hour 15 (e) Hour 16 (f) Hour 21

Figure 5.33: Visibility readings and predictions for (a) Queens on November 4th and sample images from different hours of the day (b)-(f).

the maximum discernible visibility is much shorter than the maximum ASOS value at 10.0 miles. This is also reflected in the sample images (b)-(f) where the scene looks approximately the same throughout the day. Some stations show a significant difference of performance between the training sets and the test sets. Figure 5.35a shows the visibility readings and model predictions for Manhattan on February 13th, 2022. Here the model trained on Manhattan is able to match the ASOS visibility readings closely, however, the model from the rotation where Manhattan was in the test set made predictions that did not match the ASOS readings. However, when looking at

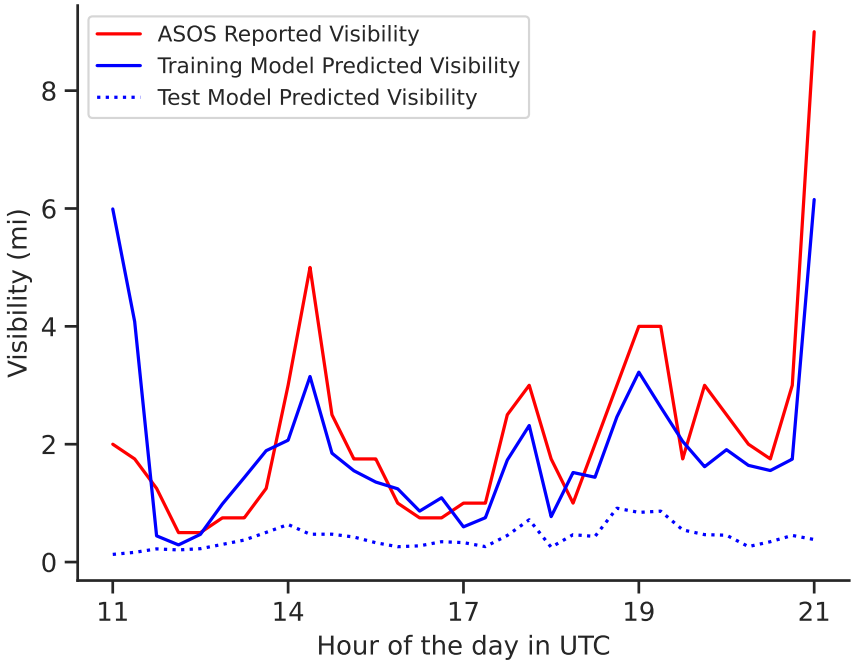


(a)

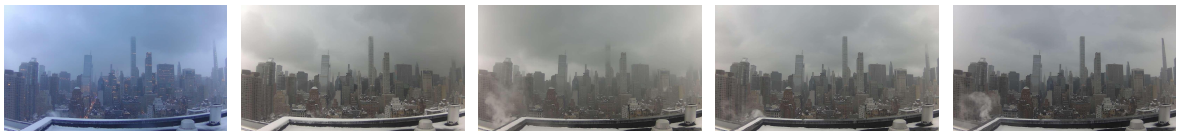
(b) 12 (c) Hour 15 (d) Hour 18 (e) Hour 20 (f) Hour 21

Figure 5.34: Visibility readings and predictions for (a) Elmira on November 11th and sample images from different hours of the day (b)-(f).

the sample images, it is difficult to tell the visibility distances represented in the images at this location, especially since the fog present is non-uniformly distributed across the scene. Overall, the models are reasonably detecting the visibility that is truly depicted within the images, even if, it does not reflect the ASOS visibility.



(a)



(b) Hour 11 (c) Hour 14 (d) Hour 17 (e) Hour 19 (f) Hour 21

Figure 5.35: Visibility readings and predictions for (a) Manhattan on February 13th and sample images from different hours of the day (b)-(f).

Chapter 6

Discussion and Conclusions

Low visibility conditions affect all types of vehicular transportation, from large operations like air travel and maritime shipping, to individual automotive vehicles. Dense fog or severe weather can greatly impact the speed, reliability, and safety of vehicular travel. While current visibility estimates are limited by the location of expensive visibility sensors, CCTV and web cameras are inexpensive and require minimal maintenance. Due to their low cost and ease of deployment, cameras have become increasingly ubiquitous in public and private spaces. Image-based visibility detection has the potential to provide accessible visibility measurements to under-supported or large areas.

However, image-based visibility estimation is not without its considerable challenges. Visibility measurements are inherently associated with the depth of a scene, however, monocular 2-D images are flat. The location of landmarks in a camera image are often unknown, and the position of these landmarks are generally not be available for new locations or sites. In addition, image-based visibility data sets difficult to procure and publicly available data sets are scarce. Existing data sets have their own drawbacks, such as images from a small number of unique locations, the lack of low visibility examples, or significant noise in the data (e.g., camera lens obstructions, sun glare, de-correlation between the images and visibility labels).

With these challenges in mind, I have proposed and described a novel solution to determine the comparative visibility from two images and translate comparisons between a unknown query image and a set of known reference images to a numerical

visibility estimate. I have also curated a new image-based visibility data set from a combination of publicly and privately available sources, ASOS network and the NYS Mesonet, respectively. This data set spans seven years and consists of ten unique locations. Since the data is collected over many years, low visibility conditions are documented in many of its different forms, at different times of day, and at different times of year. This is unique, since many visibility data sets consists of examples collected over several months or artificially generated from static scenes. Furthermore, the locations are spread across the state of New York. These locations include rural and urban landscapes that depict exceedingly different landmarks and fields of view. The scenes captured by these images also change over time, adding necessary and vital variation to the data set which aided in training robust models.

As demonstrated in Chapter 5, the model is able to determine the comparative visibility between two images with considerable accuracy, given the significant presence of mislabeled examples. The comparative approach provides tolerance for many mislabeled examples, since the visibility distances of the query and reference images do not always have to be exact. For example, a reference image may be correctly labeled with a visibility distance of 1 mile, and a query image may have a reported visibility of 2.0 miles, but a true visibility of 10.0 miles. Even though the visibility measurement is 8 miles off for the query image, the model should predict the same output label. The query image is still more visible than the reference image.

The results outlined in Section 5.1 demonstrate that the models are able to distinguish between different visibility conditions, especially in image pairs that express very different visibility conditions. However, when the query and reference images have a similar visibility, it becomes more difficult for the model to determine which image is more visible than the other. This is primarily caused by two impeding factors, 1) often times there are no visually detectable differences between certain visibility values (e.g.,

1.75 miles and 2.0 miles), and 2) there may be a de-correlation between the conditions represented in the image and the detected visibility from the ASOS sensor. For example, a reference image may be correctly labeled with a visibility distance of 1.75 miles, and a query image may have a reported visibility of 1.50 miles, but a true visibility of 2.0 miles. Even though there is only 0.5 mile error in the query image label, this flips the expected prediction. In these cases, the model may accurately measure the correct comparative visibility, but in turn cause a negative impact on its accuracy. Therefore, comparisons that are close to being equivalent are inherently difficult and are more subject to the noise in the data set.

Overall, the comparative visibility estimator shows an ability to determine the relative difference in visibility of images taken at unseen locations. Although there was a degradation in performance between the training, validation, and test sets, the model was still able to largely distinguish between different visibility conditions.

Using the comparative visibility estimator, a query image's true underlying visibility can be inferred by comparing the query to a set of labeled reference images. The output from these comparisons can be translated to a numerical visibility distance. The transition between where the model reports that the query image is more visible than a reference image and that the query image is less visible than a reference image is used as the numerical visibility prediction. Section 5.2 shows at length that this approach is able to predict reasonable visible estimates, with respect to the visual limitations present in the images.

Numerical visibility predictions from this approach tend to have a high variance. On the surface this indicates a high error, however, when looking at specific examples, I show that that there is prevalent de-correlation between what is being captured by the NYS Mesonet camera images and what is being reported by the ASOS visibility

sensors. At times, the model's predictions seem to be more accurate than the associated visibility label.

The numerical visibility results also reflect the maximum discernible visibility for many stations. Although ASOS visibility measurements report up to 10 miles, some NYS Mesonet cameras may only capture scenes that depict landmarks up to 0.5 miles and have a discernible visibility up to 2 miles. For many stations, the field of view is reduced by impeding landmarks or camera positioning. This causes the model to predict up to this maximum discernible visibility, especially in the test set results.

In addition, the model was not able to generalize to all stations. Some stations demonstrate non-uniform fog in the images, which makes it difficult to ascertain the true underlying visibility distance. In these cases, there is no accurate way to determine the true visibility for the overall scene, since multiple visibility distances are represented in one scene.

Furthermore, the model performed better on locations that were in the training set rather than in the validation and test sets. This is primarily because observational images of landscapes (especially those of the NYS Mesonet) do not to substantially change over time. In addition, low visibility conditions may look radically different based on time of day or season, however, what low visibility looks like for a specific location is fairly consistent. New years of data are unlikely to show novel low visibility conditions that are not represented in previous years. This is a key issue with much of previous work in image-based visibility estimation. Many approaches use randomized training, validation, and test sets that are not partitioned with respect to time. This causes strong autocorrelation between the different sets and limits the application of the models. The final results for these prior models do not show how the model would perform on new data, since the validation and test sets are not temporally independent from the training set. However, my approach directly addresses the temporal

autocorrelation prevalent in image-based visibility data sets by creating temporally independent training, validation, and test sets. In this work, model performance on the test set is a valid indicator on how well my approach would perform on completely new data. This approach could potentially perform even better in deployment if visibility limits, reflecting maximum discernible visibility, are specified for each site.

Image-based visibility detection for specific locations could be useful. However, this approach requires thousands of images labeled with associated visibility distances. It is exceedingly important to develop a model that can ascertain visibility at novel locations, that do not have a visibility sensor. Cross-site generalization is a task that is largely unaddressed in the vast majority of previous proposed approaches in image-based visibility estimation. All of the deep learning approaches discussed in Chapter 2 do not explicitly show results for novel or unseen locations. To my knowledge, this work is the first to explicitly attempt to construct a cross-site visibility detector and provide detailed results on the approach's generalizability.

In conclusion, this approach demonstrates an ability to determine visibility from camera images. This approach is generalizable to new locations, only requiring a small set of reference images for each location. The proposed approach also demonstrates an ability to learn in the presence of non-trivial label noise and adapt to different locations with varying fields of view. Furthermore, this work also shows an automated way to select reference images for locations where visibility estimates are accessible.

6.1 Future Work

Due to the prevalent label noise in my data set, it is vital to determine the true accuracy of the models with respect to a reliable ground truth. This can be done by collecting a clean set of human-verified labels and investigating how well the current models

perform on this cleaned data. Furthermore, the current model architecture could be trained on a hand labeled suite of images. The resulting models could possibly produce more accurate comparisons or estimations than current models trained using the ASOS labels. In addition, the current models or the proposed models, those trained with hand labeled data, could provide candidate labels or be used in another process to filter the existing labels for a second stage model. The second stage model would learn from the candidate or filtered labels in the hopes of increasing overall accuracy and generalizability beyond current results.

This work could also be extended to real-world categorical visibility labels such as the visibility grading for aviation, maritime, or automotive applications. The performance of the model could be verified by domain-experts or end users with the intent of deployment. This could prove useful for monitoring visibility over large areas with limited resources or sensor availability.

Furthermore, the current model does have some problems generalizing to new locations. Adding more stations or locations to the training set may help improve accuracy in the validation and test sets. Substantial architecture changes could also be helpful for improving the performance of the model. The overall architecture could be optimized by careful experimentation with the various hyperparamters, the input image size, and the model complexity. In addition, the model could potentially be improved by leveraging historical unlabeled data from a site and develop a site specific model that captures useful features and image regions for estimating visibility. Furthermore, integrating an attention mechanism or XAI techniques could possibly improve performance and give helpful insight on how the model currently works. Further experimentation with model explainability would allow for better communication with end users and facilitate better impacts in deployment.

Bibliography

Amani, M., Mahdavi, S., Bullock, T., and Beale, S. Automatic nighttime sea fog detection using goes-16 imagery. *Atmospheric Research*, 238:104712, 2020. ISSN 0169-8095. doi: https://doi.org/10.1016/j.atmosres.2019.104712. URL `https://www.sciencedirect.com/science/article/pii/S0169809519305447`.

American Meteorological Society. Visbility - glossary of meterology, 2012. URL `https://glossary.ametsoc.org/wiki/Visibility`. Accessed: May 31st, 2023.

Babari, R., Hautière, N., Éric Dumont, Paparoditis, N., and Misener, J. Visibility monitoring using conventional roadside cameras – emerging applications. *Transportation Research Part C: Emerging Technologies*, 22:17–28, 2012. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2011.11.012. URL `https://www.sciencedirect.com/science/article/pii/S0968090X11001641`.

Belaroussi, R. and Gruyer, D. Impact of reduced visibility from fog on traffic sign detection. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1302–1306, 2014. doi: 10.1109/IVS.2014.6856535.

Bosse, S., Maniry, D., Müller, K.-R., Wiegand, T., and Samek, W. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Transactions on Image Processing*, 27(1):206–219, 2018. doi: 10.1109/TIP.2017.2760518.

Bouhsine, T., Idbraim, S., Bouaynaya, N. C., Alfergani, H., Ouadil, K. A., and Johnson, C. C. Atmospheric visibility image-based system for instrument meteorological conditions estimation: A deep learning approach. In *2022 9th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 1–6, 2022. doi: 10.1109/WINCOM55661.2022.9966454.

Brotzge, J. A., Wang, J., Thorncroft, C. D., Joseph, E., Bain, N., Bassill, N., Farruggio, N., Freedman, J. M., Hemker, K., Johnston, D., Kane, E., McKim, S., Miller, S. D., Minder, J. R., Naple, P., Perez, S., Schwab, J. J., Schwab, M. J., and Sicker, J. A technical overview of the new york state mesonet standard ne-twork. *Journal of Atmospheric and Oceanic Technology*, 37(10):1827 – 1845, 2020. doi: https://doi.org/10.1175/JTECH-D-19-0220.1. URL `https://journals.ametsoc.org/view/journals/atot/37/10/jtech-d-19-0220.1.xml`.

Cai, B., Xu, X., Jia, K., Qing, C., and Tao, D. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016. doi: 10.1109/TIP.2016.2598681.

Carley, J. R., Matthews, M., Morris, M. T., De Pondeca, M. S., Colavito, J., Yang, R., and Toyofuku, T. Variational assimilation of web camera-derived estimates of visibility for alaska aviation. *Experimental Results*, 2:e14, 2021.

Chaabani, H., Kamoun, F., Bargaoui, H., Outay, F., and Yasar, A.-U.-H. A neural network approach to visibility range estimation under foggy weather conditions. *Procedia computer science*, 113:466–471, 2017.

Cheng, X., Liu, G., Hedman, A., Wang, K., and Li, H. Expressway visibility estimation based on image entropy and piecewise stationary time series analysis. *arXiv preprint arXiv:1804.04601*, 2018.

Cho, Y.-I. and Palvanov, A. A new machine learning algorithm for weather visibility and food recognition. *Journal of Robotics, Networking and Artificial Life*, 6:12–17, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

Edwing, R. F. NOAA's physical oceanographic real-time system (PORTS®). *Journal of Operational Oceanography*, 12(sup2):S176–S186, 2019. doi: 10.1080/1755876X.2018.1545558. URL `https://doi.org/10.1080/1755876X.2018.1545558`.

Gaudet, L. personal communication, 2021.

Hallowell, R., Matthews, M., and Pisano, P. An automated visibility detection algorithm utilizing camera imagery. In *23rd Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology (IIPS)*, 2007.

He, K., Sun, J., and Tang, X. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2011. doi: 10.1109/TPAMI.2010.168.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

Huang, G., Liu, Z., and Weinberger, K. Q. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL `http://arxiv.org/abs/1608.06993`. Accessed: June 18th, 2023.

Iowa State University, I. Asos-awos-metar data download. URL `https://mesonet.agron.iastate.edu/request/download.phtml`. Accessed: June 3rd, 2023.

Iqbal, H. Harisiqbal88/plotneuralnet v1.0.0, Dec. 2018. URL `https://doi.org/10.5281/zenodo.2526396`. Accessed: June 16th, 2023.

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Yifu, Z., Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, UnglvKitDe, Sonck, V., tkianai, yxNONG, Skalski, P., Hogan, A., Nair, D., Strobel, M., and Jain, M. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, Nov. 2022. URL https://doi.org/10.5281/zenodo.7347926. Accessed: June 8th, 2023.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Li, J., Lo, W. L., Fu, H., and Chung, H. S. H. A transfer learning method for meteorological visibility estimation based on feature fusion method. *Applied Sciences*, 11(3), 2021. ISSN 2076-3417. doi: 10.3390/app11030997. URL https://www.mdpi.com/2076-3417/11/3/997.

Li, Q., Li, Y., and Xie, B. Single image-based scene visibility estimation. *IEEE Access*, 7:24430–24439, 2019a. doi: 10.1109/ACCESS.2019.2894658.

Li, Q., Tang, S., Peng, X., and Ma, Q. A method of visibility detection based on the transfer learning. *Journal of Atmospheric and Oceanic Technology*, 36(10):1945 – 1956, 2019b. doi: https://doi.org/10.1175/JTECH-D-19-0025.1. URL https://journals.ametsoc.org/view/journals/atot/36/10/jtech-d-19-0025.1.xml.

Li, S., Fu, H., and Lo, W.-L. Meteorological visibility evaluation on webcam weather image using deep learning features. *International Journal of Computer Theory and Engineering*, 9(6):455 – 461, 2017. doi: https://doi.org/10.7763/IJCTE.2017.V9.1186. URL http://www.ijcte.org/index.php?m=content&c=index&a=show&catid=92&id=1442.

Mathew, S. and Pulugurtha, S. S. Quantifying the effect of rainfall and visibility conditions on road traffic travel time reliability. *Weather, Climate, and Society*, 14 (2):507 – 519, 2022. doi: https://doi.org/10.1175/WCAS-D-21-0053.1. URL https://journals.ametsoc.org/view/journals/wcas/14/2/WCAS-D-21-0053.1.xml.

McPherson, R. A., Fiebrich, C. A., Crawford, K. C., Kilby, J. R., Grimsley, D. L., Martinez, J. E., Basara, J. B., Illston, B. G., Morris, D. A., Kloesel, K. A., Melvin, A. D., Shrivastava, H., Wolfinbarger, J. M., Bostic, J. P., Demko, D. B., Elliott, R. L., Stadler, S. J., Carlson, J. D., and Sutherland, A. J. Statewide monitoring of the mesoscale environment: A technical update on the oklahoma mesonet. *Journal of Atmospheric and Oceanic Technology*, 24(3):301 – 321, 2007. doi: https://doi.org/10.1175/JTECH1976.1. URL https://journals.ametsoc.org/view/journals/atot/24/3/jtech1976_1.xml.

Mesonet. History of the mesonet. URL https://www.mesonet.org/about/history-of-mesonet. Accessed: May 31st, 2023.

Mi, C., Yuan, B., Ma, P., Guo, Y., Qi, L., Wang, F., Wu, W., and Wang, L. Visibility prediction based on landmark detection in foggy weather. In *2020 International Conference on Robots and Intelligent System (ICRIS)*, pages 134–137, 2020. doi: 10.1109/ICRIS52159.2020.00041.

Middleton, W. E. K. *Vision through the atmosphere*. University of Toronto Press, 1952.

Mizusawa, S., Sei, Y., Orihara, R., and Ohsuga, A. Computed tomography image reconstruction using stacked u-net. *Computerized Medical Imaging and Graphics*, 90:101920, 2021.

National Oceanic and Atmospheric Administration, F. U. United States Department of Air Force. *Automated Surface Observing System (ASOS) User's Guide*, 1998. URL https://www.weather.gov/media/asos/aum-toc.pdf. Accessed: May 31st, 2023.

National Weather Service. Automated surface observing systems, a. URL https://www.weather.gov/asos/asostech. Accessed: May 31st, 2023.

National Weather Service. Boating in fog, b. URL https://www.weather.gov/safety/fog-boating. Accessed: May 31st, 2023.

New York State Department of Motor Vehicles. Chapter 10: Special driving conditions. URL https://dmv.ny.gov/about-dmv/chapter-10-special-driving-conditions. Accessed: May 31st, 2023.

NYS Mesonet. Nys mesonet history. URL http://www.nysmesonet.org/about/history. Accessed: May 31st, 2023.

Palvanov, A. and Cho, Y. I. Visnet: Deep convolutional neural networks for forecasting atmospheric visibility. *Sensors*, 19(6):1343, 2019.

Pan, Z., Xu, J., Guo, Y., Hu, Y., and Wang, G. Deep learning segmentation and classification for urban village using a worldview satellite image based on u-net. *Remote Sensing*, 12(10):1574, 2020.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Qin, H. and Qin, H. Image-based dedicated methods of night traffic visibility estimation. *Applied Sciences*, 10(2), 2020. ISSN 2076-3417. doi: 10.3390/app10020440. URL https://www.mdpi.com/2076-3417/10/2/440.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2014.

Tarel, J.-P., Hautière, N., Caraffa, L., Cord, A., Halmaoui, H., and Gruyer, D. Vision enhancement in homogeneous and heterogeneous fog. *IEEE Intelligent Transportation Systems Magazine*, 4(2):6–20, Summer 2012a. http://perso.lcpc.fr/tarel.jean-philippe/publis/itsm12.html.

Tarel, J.-P., Hautiere, N., Caraffa, L., Cord, A., Halmaoui, H., and Gruyer, D. Vision enhancement in homogeneous and heterogeneous fog. *IEEE Intelligent Transportation Systems Magazine*, 4(2):6–20, 2012b. doi: 10.1109/MITS.2012.2189969.

Wang, M., Zhou, S., Liu, Z., and Zhang, Y. Error analysis and improved design of target board reflection for visibility measurement by the image method. *Journal of Atmospheric and Oceanic Technology*, 37(12):2299 – 2305, 2020. doi: https://doi.org/10.1175/JTECH-D-20-0057.1. URL `https://journals.ametsoc.org/view/journals/atot/37/12/jtech-d-20-0057.1.xml`.

Wauben, W. and Roth, M. Exploration of fog detection and visibility estimation from camera images. In *WMO Technical Conference on Meteorological and Environmental Instruments and Methods of Observation, CIMO TECO*, pages 1–14, 2016.

Xun, L., Zhang, H., Yan, Q., Wu, Q., and Zhang, J. Visor-net: Visibility estimation based on deep ordinal relative learning under discrete-level labels. *Sensors*, 22(16), 2022. ISSN 1424-8220. doi: 10.3390/s22166227. URL `https://www.mdpi.com/1424-8220/22/16/6227`.

Yao, S. and Huang, B. Extraction of aerosol optical extinction properties from a smartphone photograph to measure visibility. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022. doi: 10.1109/TGRS.2021.3132431.

You, Y., Lu, C., Wang, W., and Tang, C.-K. Relative cnn-rnn: Learning relative atmospheric visibility from images. *IEEE Transactions on Image Processing*, 28(1): 45–55, 2019. doi: 10.1109/TIP.2018.2857219.

Zhang, F., Yu, T., Li, Z., Wang, K., Chen, Y., Huang, Y., and Kuang, Q. Deep quantified visibility estimation for traffic image. *Atmosphere*, 14(1), 2023. ISSN 2073-4433. doi: 10.3390/atmos14010061. URL `https://www.mdpi.com/2073-4433/14/1/61`.

Zhao-zheng, C., Jia, L., and Qi-mei, C. Real-time video detection of road visibility conditions. In *2009 WRI World Congress on Computer Science and Information Engineering*, volume 5, pages 472–476, 2009. doi: 10.1109/CSIE.2009.169.

Zou, X., Wu, J., Cao, Z., Qian, Y., Zhang, S., Han, L., Liu, S., Zhang, J., and Song, Y. An atmospheric visibility grading method based on ensemble learning and stochastic weight average. *Atmosphere*, 12(7), 2021. ISSN 2073-4433. doi: 10.3390/atmos12070869. URL `https://www.mdpi.com/2073-4433/12/7/869`.

Appendix A
Implementation Details

A.1 Determining RGB vs. Grayscale Image

All NYS Mesonet camera images are stored with the same size of 720 pixels in height, 1280 pixels in width, and 3 color channels, regardless if the image was captured in RGB or grayscale modes. In order to determine which images are RGB and which were Grayscale, I compared each of the color channels. If the color channels were found to be equal I flagged the image as grayscale, and if the color channels were different I flagged the image as RGB (Gaudet, 2021).

A.2 Sunrise and Sunset Time based on Latitude and Longitude

When filtering images by sunset and sunrise time, I use the sunrise and sunset times of a location given its latitude and longitude. I use the package "suntime" in Python to do this. However, there is bug where if the sunset time for a given day is past midnight, the suntime package reports the sunset time correctly, however, reports the current day instead of the next day. This caused issues with filtering images between sunrise and sunset time, because if the bug was triggered it would flag all images as being outside of sunrise and sunset time. In order to quickly correct this, I just had a flag that if the sunset time occurs before sunrise, add one day to the sunset time.

A.3 Comparative Model Training Details

Both the CNN and DNN blocks use L2 regularization. For model training, I use an Adam optimizer and binary cross entropy for the model's loss function. Since my data set is composed of over 2 million images, the total number of possible query and reference images pairs is $(2 \cdot 10^6)^2$. This is too many pairs to iterate through for one epoch in training, so I limit the number of pairs per epoch to the total number of training images divided by two. This is the same for the validation set, where after each epoch the model is validated on a subset of the possible pairs (number of validation pairs is the total number of validation images divided by two). I also use early stopping

Hyperparamter	Value
L2 Penalty	0.0001
Learning Rate	0.0002
Early Stopping Patience	15 epochs
Early Stopping Minimum Delta	0.0001
Batch Size	256
Padding (CNN)	valid
Spatial Dropout (CNN)	0.1
Dropout (DNN)	0.3

Table A.1: Training hyperparameters.

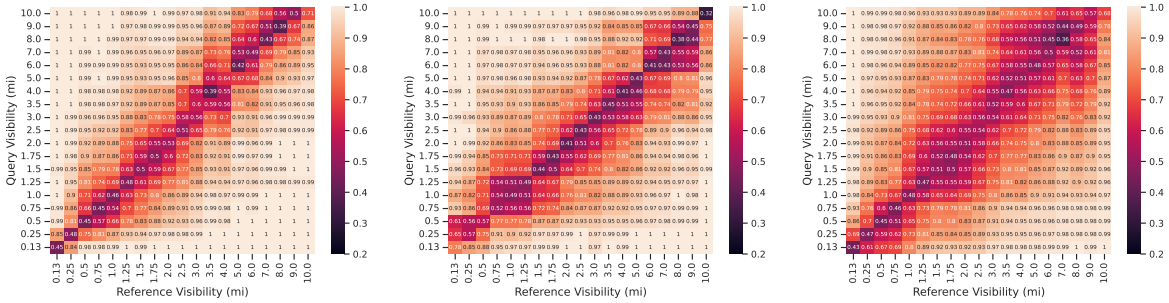
where I monitor the validation loss. The specific values of training hyperparamters can be found in Table A.1.

During training, the model weights were check-pointed after every epoch. After the training ending, the check-pointed model weights from the epoch with the highest validation accuracy were restored.

Appendix B
Detailed Results

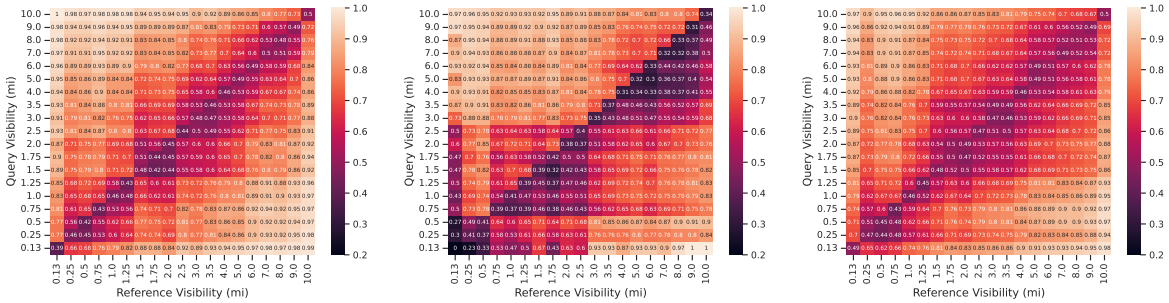
B.1 Comparative Visibility Estimator Results

These accuracy plots demonstrate the results for the comparative visibility estimator. The plots are broken down by station and year. The years 2021 and 2022 comprise the validation and test sets, respectively. The training set results are comprised of only 2019 for brevity. These plots were generated by under-sampling, such that each visibility increment has approximately the same representation in the subset. Every possible pair of images from the subset is passed through the model in order to produce the accuracies for each grid in the plots.



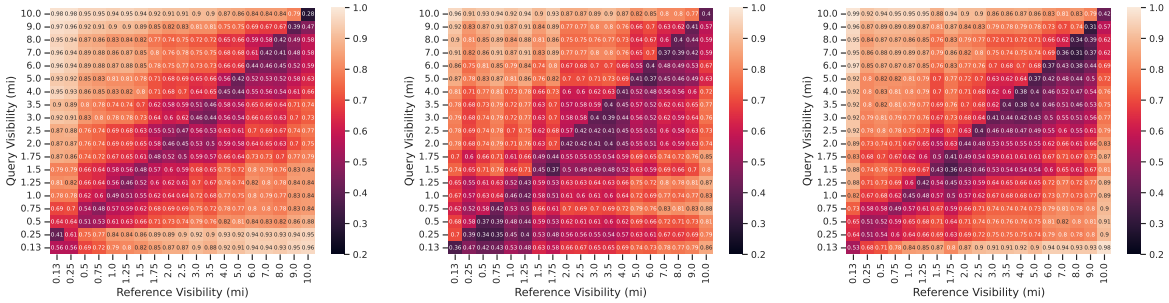
(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.1: Batavia accuracy grids across training, validation, and test years.



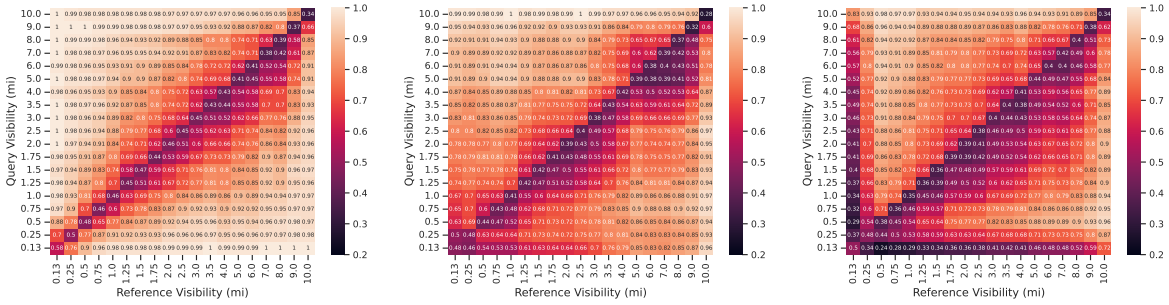
(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.2: Buffalo accuracy grids across training, validation, and test years.



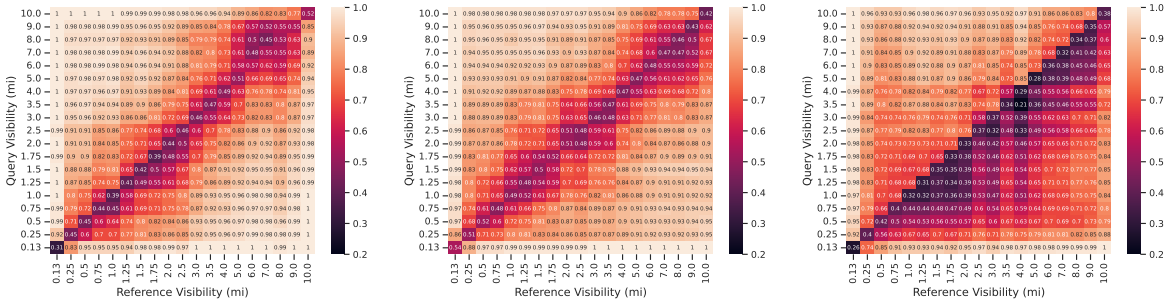
(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.3: Elmira accuracy grids across training, validation, and test years.



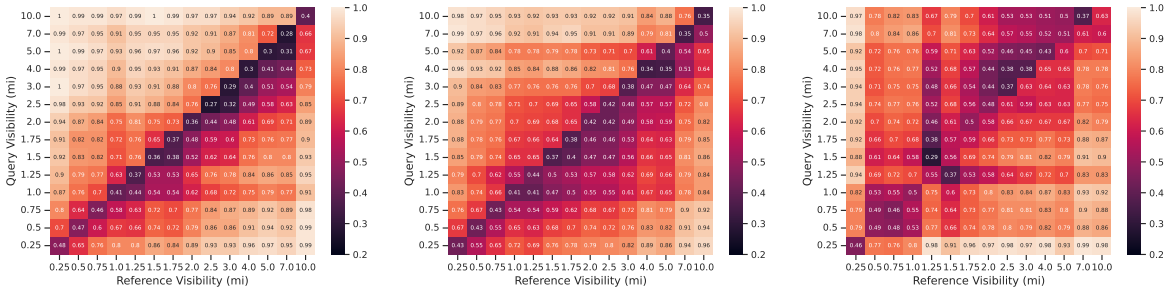
(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.4: Gabriels accuracy grids across training, validation, and test years.



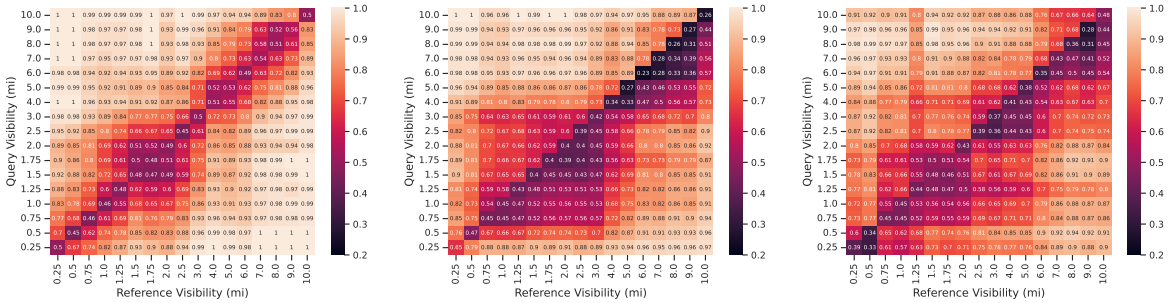
(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.5: Glen Falls accuracy grids across training, validation, and test years.



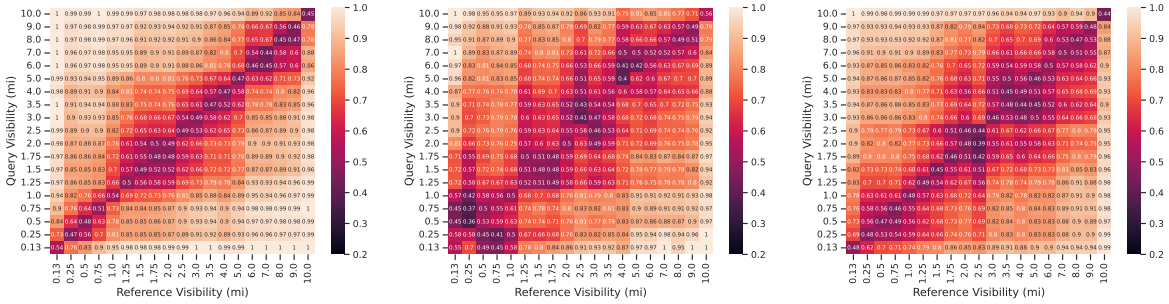
(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.6: Johnstown accuracy grids across training, validation, and test years.



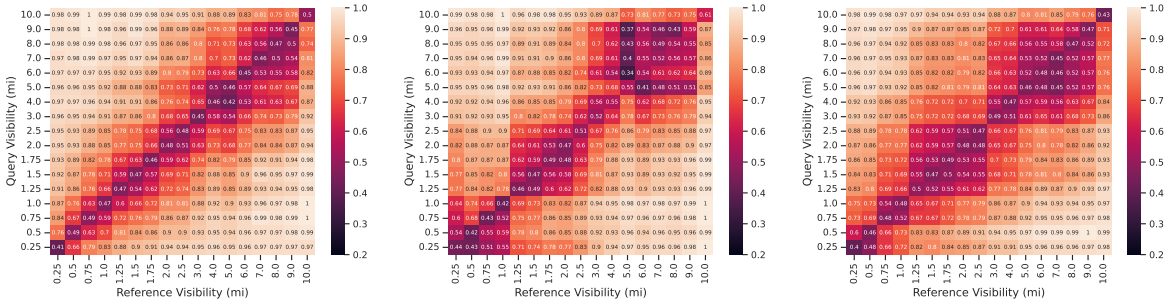
(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.7: Manhattan accuracy grids across training, validation, and test years.



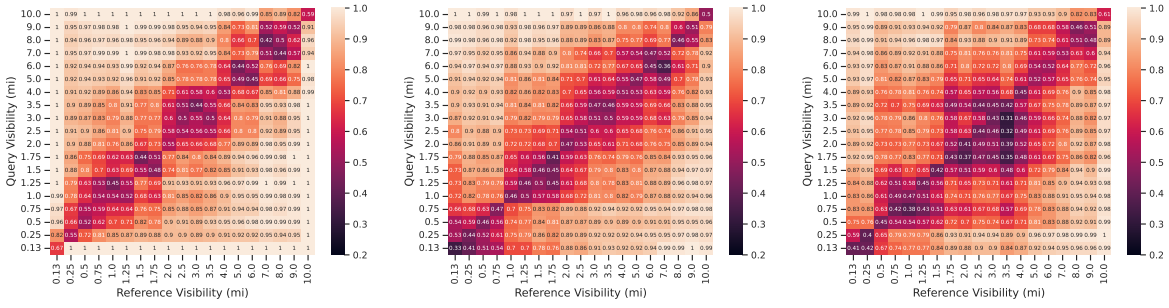
(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.8: Penn Yan accuracy grids across training, validation, and test years.



(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.9: Potsdam accuracy grids across training, validation, and test years.



(a) Training Set, 2019 (b) Validation Set, 2021 (c) Test Set, 2022

Figure B.10: Queens accuracy grids across training, validation, and test years.

B.2 Numerical Visibility RMSE Results

These tables depict the root mean squared error (RMSE) for the numerical visibility estimation task. The tables break down the error by station and by visibility increment. Some stations did not have all visibility increments represented in every year. A horizontal line is used to denote station and visibility increment combinations that did not occur.

Visibility (mi)	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
0.13	0.51	1.12	0.99	1.24	0.22	—	—	0.61	—	1.43
0.25	0.68	0.77	1.1	0.96	0.79	1.1	1.02	1.61	0.8	1.4
0.5	1.01	0.87	0.95	1.11	0.82	1.3	1.2	1.73	0.6	1.89
0.75	1.05	0.92	0.93	1.24	0.74	0.75	1.8	1.6	0.67	1.51
1.0	1.03	0.78	1.13	1.37	0.73	1.13	1.44	1.59	0.53	1.64
1.25	1.11	0.62	0.61	1.63	0.86	1.5	2.35	2.13	0.85	0.87
1.5	1.43	0.71	0.5	1.23	0.57	1.15	1.45	1.7	0.66	1.19
1.75	1.67	0.79	0.52	1.45	0.57	0.64	1.81	1.65	0.55	0.97
2.0	1.53	0.68	0.58	1.35	0.73	0.6	1.68	1.71	0.59	1.42
2.5	1.78	1.29	0.83	1.39	0.88	1.1	1.73	1.83	0.8	1.39
3.0	1.78	1.33	1.15	1.58	1.13	1.51	1.82	1.64	1.13	1.7
3.5	1.7	1.62	1.6	1.75	1.41	—	—	1.5	—	1.39
4.0	1.75	2.04	2.06	1.96	1.75	2.17	2.04	1.58	1.76	1.84
5.0	1.76	2.9	2.96	2.33	2.34	2.93	2.18	1.8	2.44	2.04
6.0	1.75	3.79	3.95	2.82	3.22	—	2.15	2.13	3.31	2.29
7.0	1.75	4.58	4.91	3.34	3.99	4.47	2.51	2.61	4.33	2.61
8.0	1.84	5.42	5.88	3.83	4.8	—	2.77	3.25	5.1	2.6
9.0	1.9	5.91	6.88	3.96	5.57	—	3.69	3.99	5.84	3.09
10.0	1.21	6.31	7.59	3.1	4.84	5.6	2.8	3.52	5.41	1.59

Table B.1: Average root mean square error by visibility distance for each station when it was represented in the training set, but using an unseen year of data from 2021.

Visibility (mi)	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
0.13	0.34	0.66	0.68	1.93	0.41	—	—	0.96	—	1.88
0.25	1.09	1.19	0.93	1.74	1.06	1.05	0.78	1.6	1.06	1.34
0.5	1.06	0.73	0.88	1.17	1.6	2.03	0.95	1.48	0.6	2.36
0.75	0.92	0.58	0.78	1.08	1.6	1.58	2.05	1.46	1.0	1.43
1.0	0.94	0.92	0.69	1.87	0.91	0.58	0.9	1.44	1.11	1.61
1.25	1.22	0.57	0.62	2.69	1.06	0.76	2.07	1.47	0.62	1.48
1.5	1.07	1.31	0.61	2.2	0.99	1.11	1.02	1.86	0.98	1.64
1.75	1.05	1.35	0.65	2.14	1.26	0.78	1.55	1.77	1.02	2.15
2.0	1.26	0.9	0.59	1.88	1.0	0.92	1.5	1.91	0.94	1.87
2.5	1.23	1.23	0.84	2.57	1.17	1.03	1.36	1.82	1.04	1.72
3.0	1.53	1.46	1.2	2.11	1.31	2.0	1.51	2.03	1.7	1.69
3.5	1.74	1.75	1.67	2.4	1.53	—	—	2.04	—	1.75
4.0	1.83	2.16	2.18	2.07	1.84	1.93	2.14	2.06	1.83	1.86
5.0	2.12	3.1	3.11	2.43	2.58	2.64	2.71	2.25	2.62	2.14
6.0	2.4	4.01	4.03	2.8	3.35	—	3.06	2.65	3.3	2.23
7.0	2.7	4.98	5.05	3.14	4.15	4.03	3.59	3.13	4.19	2.48
8.0	3.38	5.78	6.04	3.48	4.86	—	3.99	3.8	5.09	3.07
9.0	3.54	6.85	6.99	3.97	5.66	—	4.79	4.22	5.88	4.02
10.0	2.69	6.77	7.62	2.56	4.59	5.05	4.38	3.13	5.28	2.1

Table B.2: Average root mean square error by visibility distance for each station when it was represented in the training set, but using an unseen year of data from 2022.

Visibility (mi)	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
0.13	0.21	0.19	1.75	4.59	0.29	—	—	0.94	—	1.66
0.25	0.27	3.72	1.89	4.32	2.31	0.48	2.24	2.67	1.61	2.02
0.5	0.45	3.21	1.58	3.44	2.53	0.39	1.05	2.78	1.41	3.01
0.75	0.67	4.55	1.44	2.97	2.97	0.28	2.65	2.59	0.99	2.86
1.0	0.98	4.18	1.55	2.9	2.77	0.26	2.48	2.53	1.05	3.04
1.25	1.07	3.58	1.15	3.5	2.71	1.2	3.14	3.17	1.1	2.58
1.5	1.34	4.8	0.93	3.21	2.45	0.57	2.96	2.74	1.0	2.8
1.75	1.64	4.45	0.71	3.5	2.51	0.73	2.73	2.78	1.04	2.93
2.0	1.39	4.31	0.64	3.33	3.02	0.94	3.22	3.06	1.09	3.54
2.5	2.04	4.62	0.59	3.37	2.66	1.4	3.16	2.96	1.08	3.18
3.0	2.22	4.7	0.82	3.73	3.21	1.9	3.56	2.63	1.45	3.62
3.5	2.34	4.2	1.29	3.67	3.18	—	—	2.38	—	3.12
4.0	2.76	4.0	1.68	3.52	3.53	2.82	3.99	2.47	1.7	3.43
5.0	3.23	3.38	2.61	3.35	3.61	3.84	4.22	2.23	2.1	3.25
6.0	3.36	2.89	3.57	2.92	3.59	—	4.06	2.04	3.07	2.81
7.0	3.66	2.41	4.47	2.8	3.48	5.77	4.58	2.11	3.73	2.26
8.0	3.95	1.97	5.39	2.79	3.47	—	4.91	2.57	4.55	2.03
9.0	4.5	1.72	6.37	2.99	3.77	—	5.73	3.22	5.1	1.92
10.0	2.11	2.8	6.41	1.53	3.19	8.66	4.46	2.58	5.34	1.14

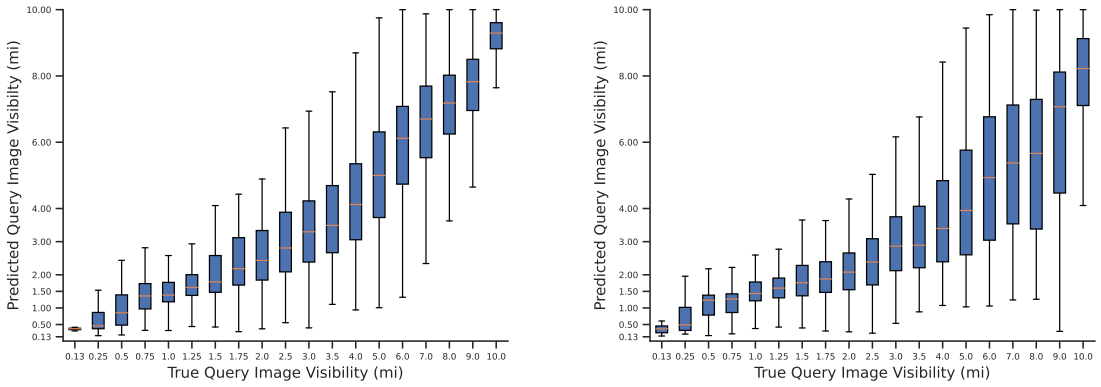
Table B.3: Average root mean square error by visibility distance for each station when it was not represented in the training set, but rather in the validation set. In addition, this uses validation data from the year 2021.

Visibility (mi)	BATA	BUFF	ELMI	GABR	GFAL	JOHN	MANH	PENN	POTS	QUEE
0.13	0.11	1.36	0.89	7.89	1.22	—	—	1.67	—	2.77
0.25	0.25	2.29	1.18	5.77	3.8	0.37	0.09	2.92	0.7	2.4
0.5	0.24	1.85	1.01	3.53	4.95	0.33	0.26	2.44	0.35	3.9
0.75	0.36	1.55	0.9	3.38	4.44	0.18	0.47	2.06	0.55	3.17
1.0	0.49	2.37	0.71	4.1	3.72	0.3	0.68	1.74	1.13	3.68
1.25	0.63	2.31	0.75	4.83	4.22	0.42	1.71	1.62	0.54	3.46
1.5	0.78	3.82	0.83	4.93	3.84	0.71	1.18	2.1	0.78	3.58
1.75	0.98	3.31	0.79	5.06	3.71	0.87	1.41	2.2	1.19	4.28
2.0	1.17	3.42	0.8	5.17	4.14	1.07	1.77	2.43	1.13	3.78
2.5	1.62	3.26	1.0	5.25	4.38	1.53	2.09	2.31	1.45	3.98
3.0	2.09	3.29	1.35	4.93	4.28	2.34	2.52	2.48	1.98	3.62
3.5	2.54	2.84	1.84	4.62	4.32	—	—	2.34	—	2.88
4.0	2.91	2.99	2.33	4.32	4.03	3.07	3.49	2.41	2.27	3.25
5.0	3.8	2.83	3.24	3.83	3.77	4.01	4.45	2.32	3.05	2.8
6.0	4.59	2.65	4.2	3.27	3.21	—	5.34	2.52	3.79	2.33
7.0	5.44	2.96	5.21	2.73	2.76	5.88	6.28	2.68	4.82	1.84
8.0	6.42	3.32	6.11	2.12	2.31	—	7.27	3.26	5.72	1.56
9.0	7.0	4.19	7.12	1.75	1.77	—	8.31	3.54	6.58	1.96
10.0	6.89	3.25	7.47	0.79	1.1	8.6	9.15	2.29	6.01	1.35

Table B.4: Average root mean square error by visibility distance for each station when it was not represented in the training set, but rather in the test set. In addition, this uses test data from the year 2022.

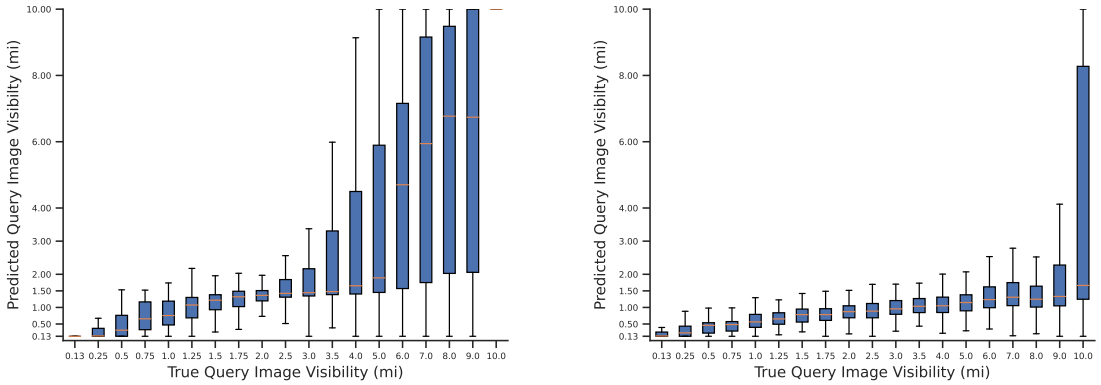
B.3 Numerical Prediction Results Break Down

These boxplots demonstrate the numerical visibility prediction results broken down by station and year. The x-axis bins the images by their reported ASOS visibility increment. The y-axis shows the spread of numerical visibility predictions for each visibility increment. The 25th percentile, median, and 75th percentile are show in the boxplots in the standard way, however, the whiskers represent 1.5 times the interquartile range. This was done to omit the numerous outliers that would have cluttered the boxplots (conveniently, this is the default behavior of matplotlib).



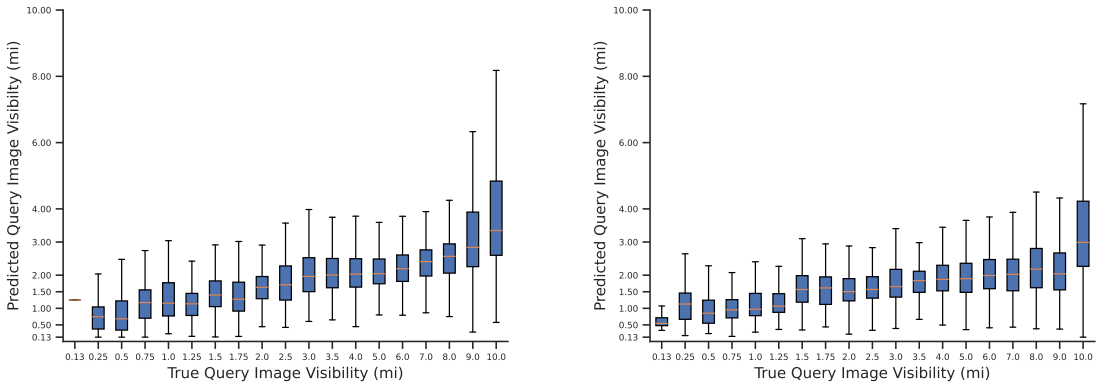
(a) (b)

Figure B.11: Boxplots for numerical predictions on Batavia 2021 and 2022 using a model that was trained with Batavia in the training set.



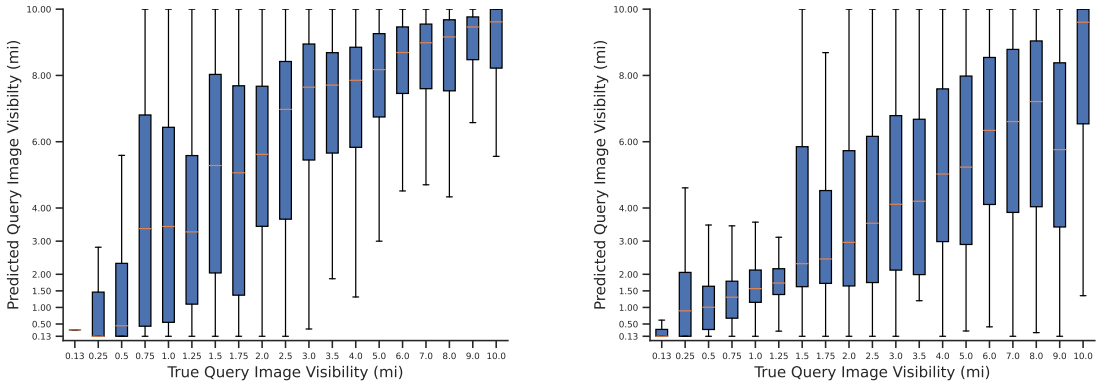
(a) (b)

Figure B.12: Boxplots for numerical predictions on the validation set Batavia 2021 and the test set Batavia 2022.



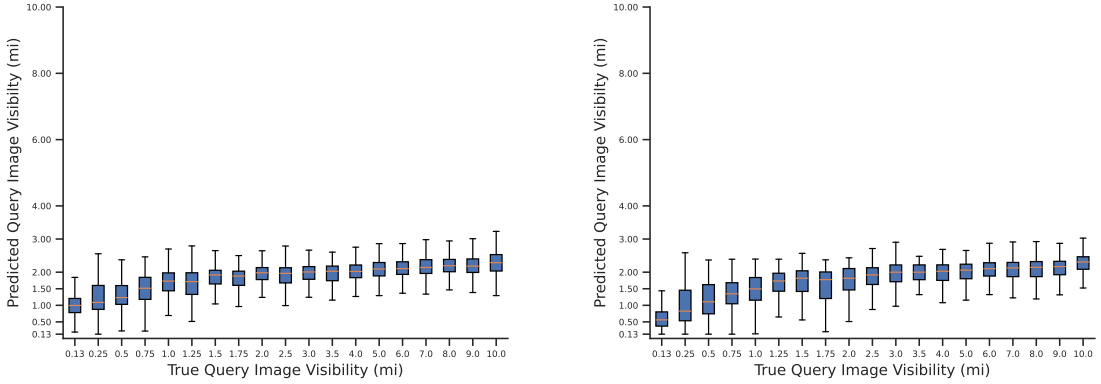
(a) (b)

Figure B.13: Boxplots for numerical predictions on Buffalo 2021 and 2022 using a model that was trained with Buffalo in the training set.



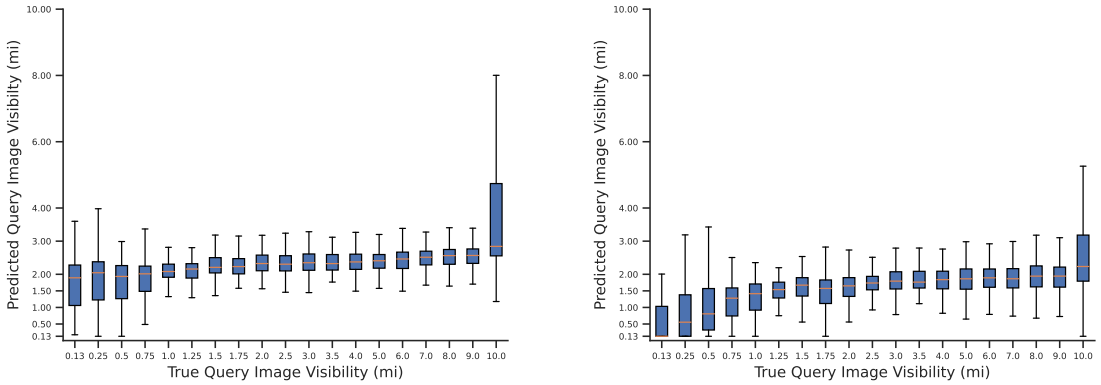
(a) (b)

Figure B.14: Boxplots for numerical predictions on the validation set Buffalo 2021 and the test set Buffalo 2022.



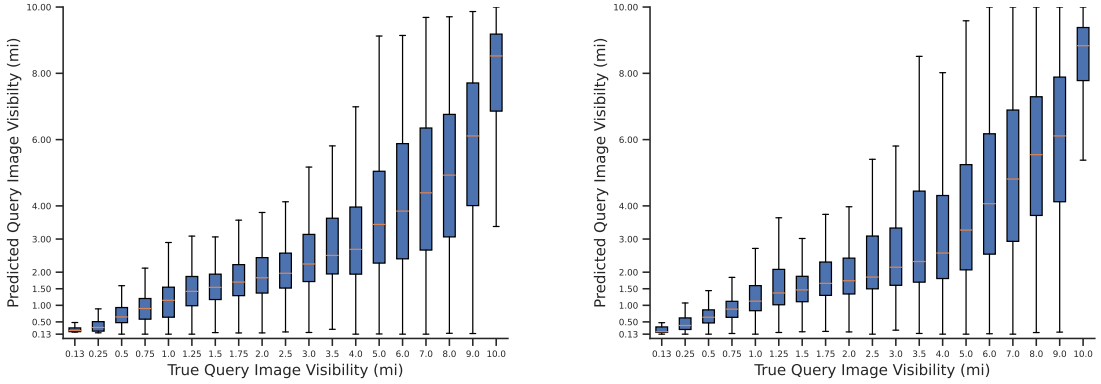
(a) (b)

Figure B.15: Boxplots for numerical predictions on Elmira 2021 and 2022 using a model that was trained with Elmira in the training set.



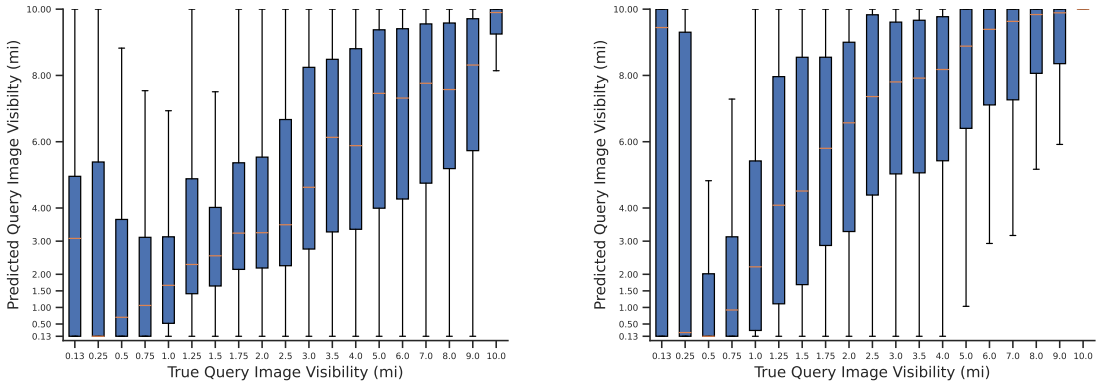
(a) (b)

Figure B.16: Boxplots for numerical predictions on the validation set Elmira 2021 and the test set Elmira 2022.



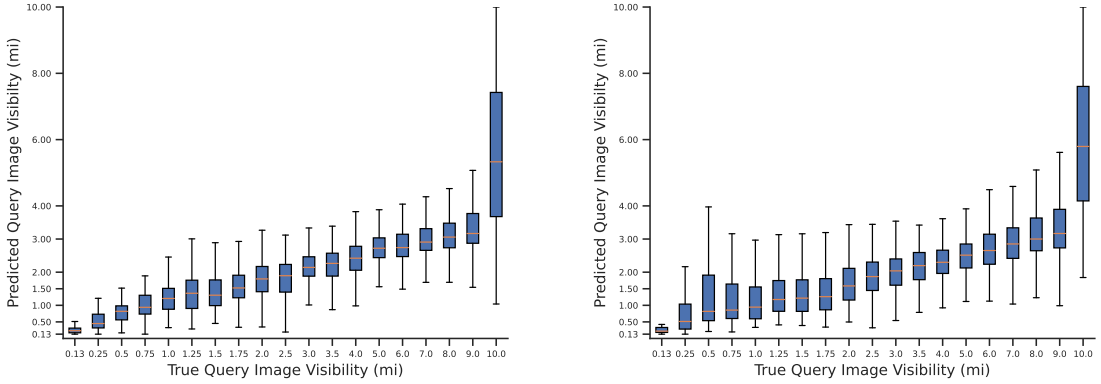
(a) (b)

Figure B.17: Boxplots for numerical predictions on Gabriels 2021 and 2022 using a model that was trained with Gabriels in the training set.



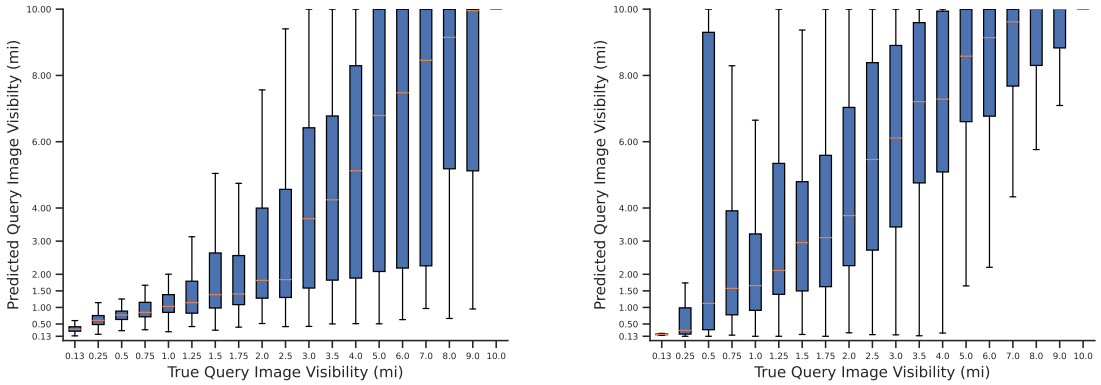
(a) (b)

Figure B.18: Boxplots for numerical predictions on the validation set Gabriels 2021 and the test set Gabriels 2022.



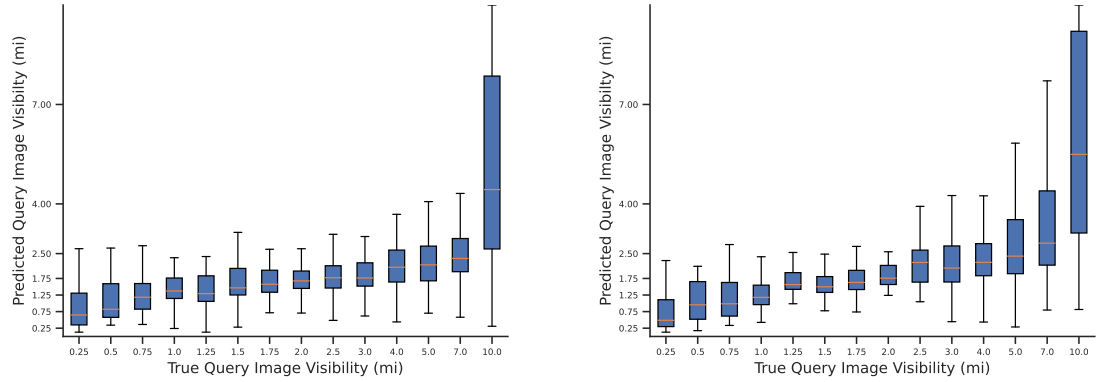
(a) (b)

Figure B.19: Boxplots for numerical predictions on Glen Falls 2021 and 2022 using a model that was trained with Glen Falls in the training set.



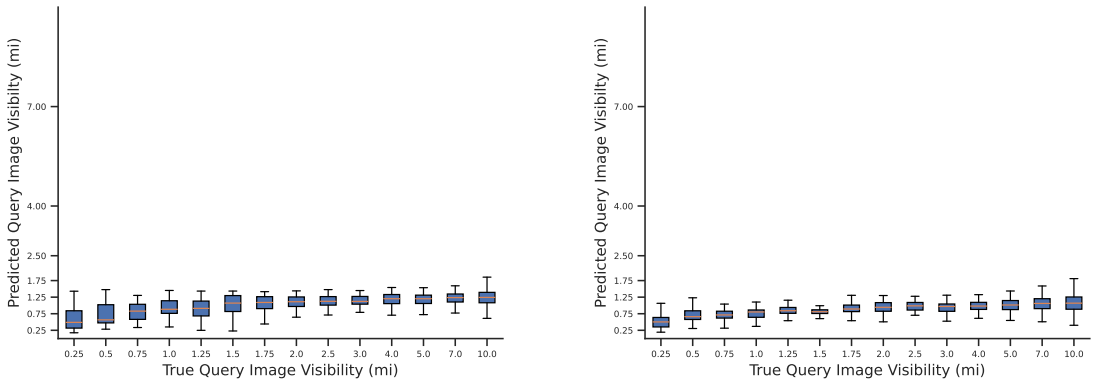
(a) (b)

Figure B.20: Boxplots for numerical predictions on the validation set Glen Falls 2021 and the test set Glen Falls 2022.



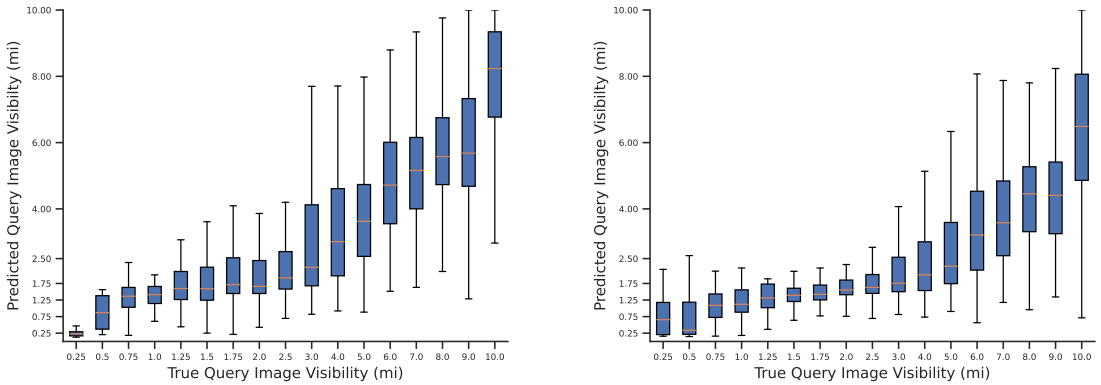
(a) (b)

Figure B.21: Boxplots for numerical predictions on Johnstown 2021 and 2022 using a model that was trained with Johnstown in the training set.



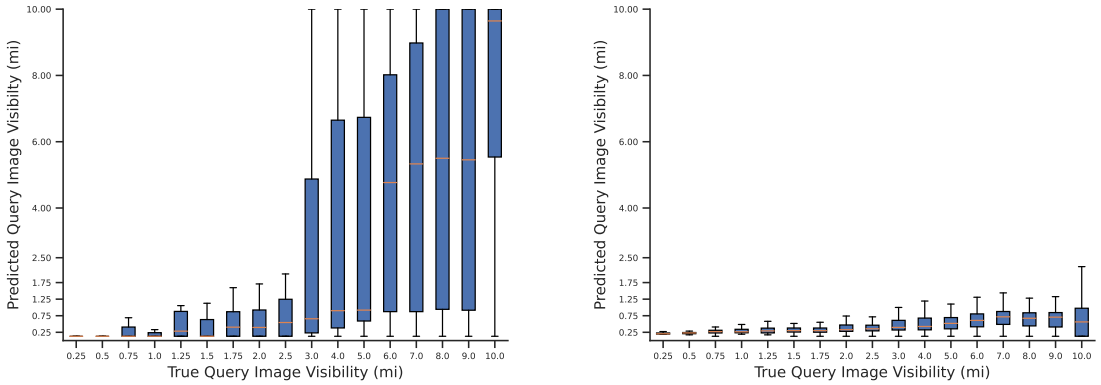
(a) (b)

Figure B.22: Boxplots for numerical predictions on the validation set Johnstown 2021 and the test set Johnstown 2022.



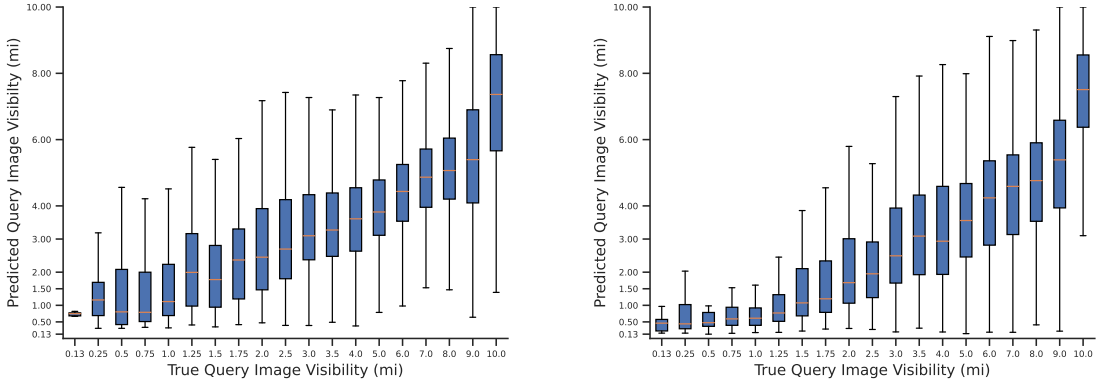
(a) (b)

Figure B.23: Boxplots for numerical predictions on Manhattan 2021 and 2022 using a model that was trained with Manhattan in the training set.



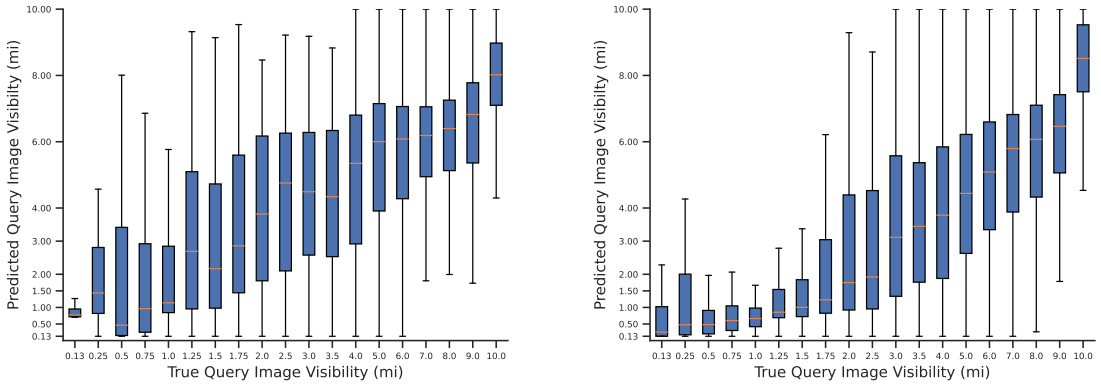
(a) (b)

Figure B.24: Boxplots for numerical predictions on the validation set Manhattan 2021 and the test set Manhattan 2022.



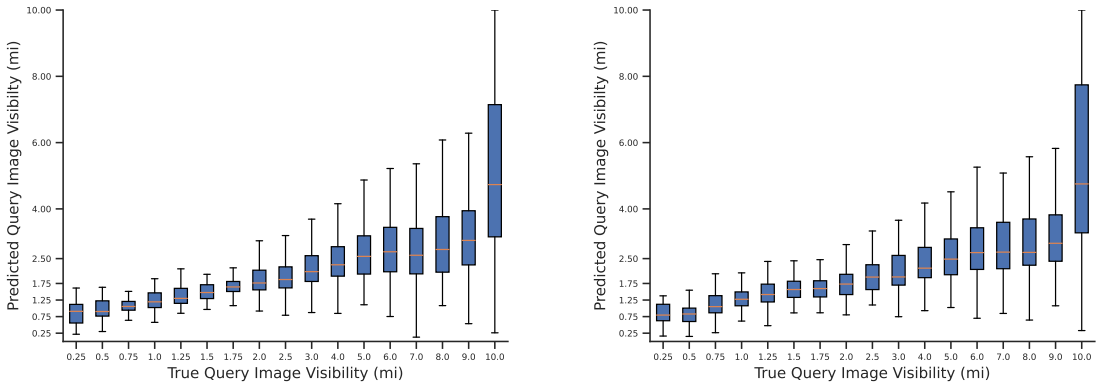
(a) (b)

Figure B.25: Boxplots for numerical predictions on Penn Yan 2021 and 2022 using a model that was trained with Penn Yan in the training set.



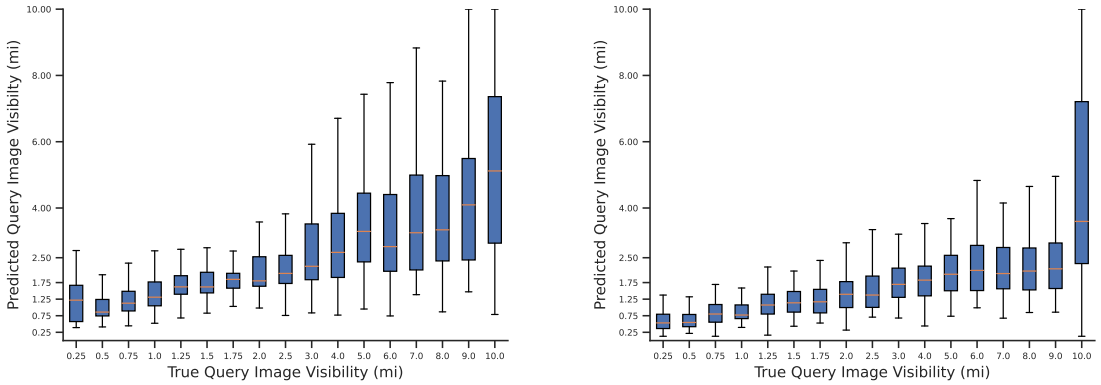
(a) (b)

Figure B.26: Boxplots for numerical predictions on the validation set Penn Yan 2021 and the test set Penn Yan 2022.



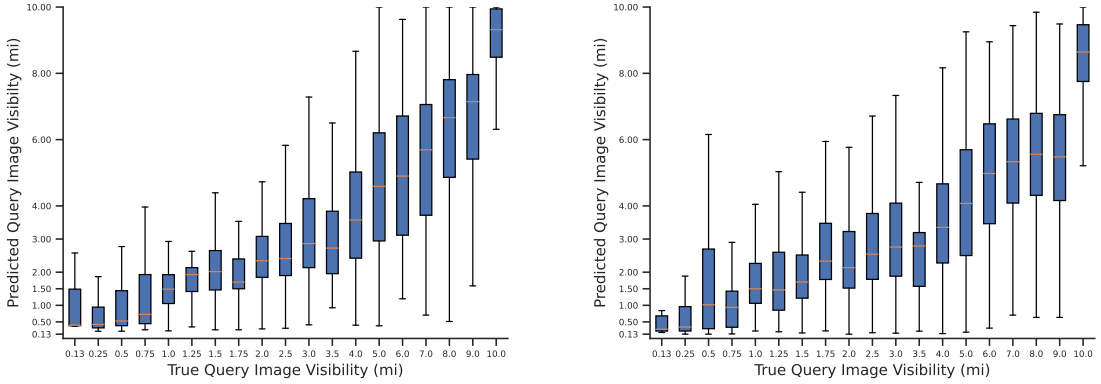
(a) (b)

Figure B.27: Boxplots for numerical predictions on Potsdam 2021 and 2022 using a model that was trained with Potsdam in the training set.



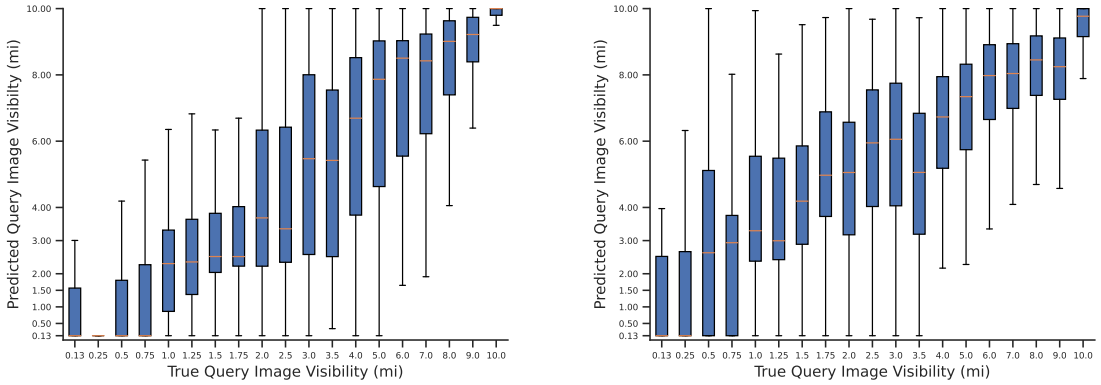
(a) (b)

Figure B.28: Boxplots for numerical predictions on the validation set Potsdam 2021 and the test set Potsdam 2022.



(a) (b)

Figure B.29: Boxplots for numerical predictions on Queens 2021 and 2022 using a model that was trained with Queens in the training set.



(a) (b)

Figure B.30: Boxplots for numerical predictions on the validation set Queens 2021 and the test set Queens 2022.