



**HAL**  
open science

## Learning to compare visibility on webcam images

Pierre Lepetit, Cécile Mallet, Laurent Barthès, Nicolas Viltard

► **To cite this version:**

Pierre Lepetit, Cécile Mallet, Laurent Barthès, Nicolas Viltard. Learning to compare visibility on webcam images. Climate Informatics 2020. 10th International Conference on Climate Informatics, Sep 2020, Oxford, United Kingdom. hal-02926386v2

**HAL Id: hal-02926386**

**<https://hal.science/hal-02926386v2>**

Submitted on 3 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# LEARNING TO COMPARE VISIBILITY ON WEBCAM IMAGES

Pierre Lepetit<sup>1,2</sup>, Cécile Mallet<sup>1</sup>, Laurent Barthes<sup>1</sup>, Nicolas Viltard<sup>1</sup>

**Abstract**—Since the 2000’s, webcams are considered as an interesting source of opportunistic meteorological data. This short study deals with the comparison of visibility, in a meteorological sense, between images. A new dataset has been built from publicly available webcam sequences. An original labeling process, based on a mergesort algorithm, allowed us to sort more than 400 webcam sequences with respect to visibility. Standard Convolutional Neural Networks have been trained to predict pairwise comparisons and tested on independent webcams that are colocalized with visibilimeters. Results on the comparison task are promising.

## I. INTRODUCTION

Despite the proliferation of observations, the accurate monitoring of surface parameters such as the meteorological visibility remains a challenge. The latter is defined as the greatest distance at which an object can be recognized [1].

On the one hand, forecast of visibility from large scale teledetection measurements is hard to achieve and this parameter is too local to be extrapolated from the weather stations, where it is instrumentally or manually measured. On the other hand, a better mapping of visibility reduction would enable interesting developments in intelligent transport system [2].

To complement the available measures, automatic processing of images from the ubiquitous road webcams appeared to be a promising idea [3], [4]. Over the past 20 years, this topic has been studied in depth with an increasing proportion of data-driven methods [2], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. However, there is few work (e.g. [15]) on the issue of the inter-scene generalization, e.g. when the predictive model is tested on scenes that have not been seen during the training phase.

Recently, deep learning appeared well suited to tackle the inter-scene generalization problem [15]. But a supervised approach of that problem is limited by the fact that reliable data sets with numerous different scenes are still missing.

First, if the webcam archives are abundant, images with low-visibility are rare. Our first contribution was to gather a large data set (AMOSvv) of 17,961 images parted in 426 webcam sequences rich in low-visibility events. For the most part, our sequences have been extracted from the AMOS archives [16]. The extraction windows span around snowfall events, when visibility varies widely [17]. These images present several difficulties for an automatic processing. For example, the lighting variations, the changing look of roads, grounds, trees and roofs due to the wetting and the settling of the snow, the frequent droplets and snowflakes deposited on the protection lens (see Fig.1 and Table 2). These effects weaken the conventional approaches based on mean contrast or edge detection.

Second, another problem comes from the lack of images associated with reliable visibility measurements. To our knowledge, the publicly-available data sets with instrumentally-derived visibility only contain one or few scenes. The remaining alternative relies on handcrafted labeling, as it is done by You et al. [15]. These authors developed a ranking model that ranks single images with respect to their apparent visibility. This model is trained on a set of manually ordered pairs of images. Pairwise labeling has been chosen because, without any knowledge on the scenes, a human annotator could hardly do better than comparing the visibility on two different images. Our approach relies on the same idea. However, comparisons were made on pairs of images of a same sequence. Moreover, this work was done by a graduate in meteorology. This ensured fine-grained, homogeneous and rigorous comparisons, whereas naive inter-scene comparisons are necessarily rougher

Corresponding author: P.Lepetit, pierre.lepetit@latmos.ipsl.fr  
<sup>1</sup>LATMOS, Guyancourt, France <sup>2</sup>CNRM, Toulouse, France

fig 1.a : examples of scenes of the training, validation and test sets



fig 1.b : five scenes of the TENEBRE network



Fig. 1. 1.a: examples of scenes of the AMOSvV data set. 1.b: from left to right: the sequences 1-5 of the TENEBRE network.

and may suffer from misleading impressions.

In practice, these comparisons were made following a quick labeling process where transitivity and pairs of incomparable images (termed “incomparable pairs”) are taken into account. This labeling process is based on the poset-mergesort algorithm of Daskalakis et al. [18]. It has been used to complete the day time part of our dataset AMOSvV with 195.000 ordinal labels on visibility, comprising strict-ordered pairs and incomparable pairs.

In this paper, we also present baseline performances for the comparison task on independent webcams of the TENEBRE network (Météo-France). Instrumental labels are used to test standard deep learning architectures. The use of incomparable pairs during the training has been addressed.

In the section 2, we quickly present the related works. Section 3 is devoted to the collection of the webcam sequences. Our labeling process is explained in section 4. The learning framework and the first results are presented in section 5.

## II. RELATED WORK

### A. Estimation of meteorological visibility

Image-based estimation of meteorological visibility appeared twenty years ago. Early works showed that fixed camera with controlled settings were usable to compute a proxy for visibility estimation [3], [4]. In the most part of subsequent studies, the train and test images are coming from the same device [19], [9], [7],

[10], [11], [20], [8]. Visibility estimations have first been obtained thanks to specific descriptors, as strength of detected contours [21], color distribution [22] or contrasts [2]; these descriptors often being motivated by a physical model [4], [23]. Progressively, estimation methods have been based on generic machine learning algorithms as support vector machine (SVM) [9], [10], [11] and Convolutional Neural Networks (CNN) [12], [13], [14].

Manual estimation of absolute visibility has been practiced in [9]. The authors use their knowledge of the scene to manually classify meteorological visibility into bins of variable sizes. They point out frequent discrepancies between the instrumentally-derived visibility and the manually derived visibility. In our test set, similar discrepancies have been observed but the global concordance was good (see section 4).

### B. Relative estimation

Parikh et al. [24] develop the idea that some attributes are incompatible with a handcrafted categorical classification but could give rise to ordinal comparisons. They propose to learn these “relative” attributes in a standard learning to rank framework [25]. Zoran et al. [26] used ordinal relationships between superpixels to predict depth and intrinsic image decomposition. They show that a learning based on manually labeled comparisons conducts to competitive results for these both middle-vision tasks. Their framework contains two main steps. They first train a deep classifier to predict an intra-image

set of ordinal relations. For a new image, predicted orders between pixels are then translated into a set of weighted quadratic and linear constraints, the solving of which form the second step. Conversely to this previous works, the present one is focused on a comparison task.

Ordinal labeling as a basis for weakly-supervised learning has also been used for visibility estimation [15]. In this study, ordinal labels are used in the same learning to rank framework as Parikh et al. Their modular model takes one image as input. Its output is a real value. It is trained on manually ordered pairs such that miss-ordered outputs are penalized.

Conversely, the images of our training pairs are stacked before the forward phase (“6-channels” inputs). A CNN is then trained to predict a partial order, as in [26]. In accordance to other studies [27], we indeed suppose that prediction from stacked images will be easier because the model has not to build its own absolute scale of visibility.

A last difference holds in our training pairs, which always contain two images of the same scene.

In the works of Zoran et al. [26] and You et al. [15], incomparability is implicitly thought as relation of proximity between the targeted values. For example, if the predictions made on incomparable inputs widely differ, the model is penalized (or bad-scored, in [15]). But in our data set, incomparable images could also be associated with very different values of visibility. Indeed, incomparability is mostly due to corrupted images or dramatic changes in the lighting. From a more theoretical point of view, Cheng et al. observed that dealing with “true partial order” may call for different learning algorithms [28]. In this preliminary work, we simply took incomparability as a third class to predict.

### III. COLLECTION OF WEBCAM SEQUENCES

To build our training and validation data sets, two sources of webcam sequences have been used: the webcams of the french DIR (Direction Interdépartementale des Routes) and the AMOS [16] archives. The AMOS archives were the main source. The way our images have been extracted is described in the following paragraph. The AMOS webcam archives are divided into directories. Each directory contains day and night time images of one or several webcams. Examples of five AMOS scenes are given in Fig.1.a.

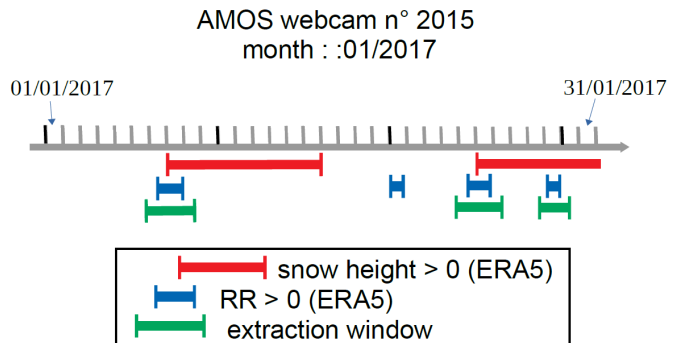


Fig. 2. Definition of the extraction windows around snowfall events.

We first have selected 500 AMOS webcams. These webcams are variously parameterized: the height with respect to the ground, the tilt angle, the color depth, the image size, differ widely. But the selected scenes always contain a road. This restriction has two interesting advantages: it reduces the input diversity while keeping all its interest for the main practical application (e.g. in road meteorology). Second, the presence of a road allows a better three-dimensional comprehension of the image, especially from the vanishing lines of the road edges.

All kind of road (see Fig.1.a and Table 1) have been selected but mountain roads. Actually, on mountain images, the variations of the cloud ceiling complicates the manual estimation of visibility.

On that first corpus, the median period of the sampling is 10 minutes and the complete sequences generally span over several years. But they contain only few events of low-visibility.

Meteorological data helped us to define extraction windows. From the ERA-5 reanalysis [29] we downloaded gridded data centered on the locations of the AMOS webcams. We used the snowfall rate and the snow height parameters. Indeed snowfalls are known to strongly affect the meteorological visibility [17]. These data were reduced to scalar series and thresholded to define periods of non-zero snow height (red intervals of Fig.2) and periods of non-zero snowfall rates (blue intervals). Our extraction windows has been taken around the periods of snowfall that initiate snow covers (green intervals). Margins of 18 hours were applied to also sample images with greater visibility.

After that, in each sequence, about 50% of the images were rejected to avoid a strong redundancy. The properties of our final training and validation data sets are summarized in Table 2. The 66 sequences of the validation set contain roughly the same proportion of type of scenes as the training set.

Our test sets are made with the archives of TENEBRE network (Météo-France). These archives span over 9 months of winter (2011–2012, 2012–2013 and 2017–2018) with a sampling period of 10 minutes. The webcams of this network are hosted in weather stations and colocalized with DF320 visibilimeters (Degreane-Horizon). In this work, we only use five TENEBRE webcams (see Fig. 1.b). Note that, on the five scenes, the only one that clearly matches with the training domain is the second one. The others are atypical: there is no road (scenes 1 and 3), or the tilt angle is anormal (scenes 4 and 5). Scene 4 is particularly challenging because there is no background.

From the archives of these five webcams, images of all kind of weather were selected to form the TENEBREq data set. All the images with low visibilities ( $< 1000 m$ ) were kept, as well as the images taken during precipitation events and those with settling snow. Sequences of TENEBREq were completed by around 1,000 images of good weather chosen at random.

The sizes of these five sequences after selection are precised in Table 3. The scores of the section 5 were computed on day time images of TENEBREq.

#### IV. LABELS

Our labeling process counts two steps. During the first step (A.), the images of each sequence are browsed in the chronological order. The annotator is invited to give two kinds of labels : a weather class and an ordinal label that is relative to the current image and the preceding one. During the second step (B.), we extend the ordinal labeling to non-consecutive images following a merge sort algorithm. These two steps are detailed further.

For each sequence  $i$ , these steps allow to build a pair of graphs  $(dg_i, ug_i)$  that will be subsequently used during the training. Both graphs share the same nodes. These nodes are the images of the  $i^{th}$  sequence.  $dg_i$  is a directed graph and its edges represent all the strictly ordered pairs.  $ug_i$  is the undirected graph of

incomparable pairs. Ideally, at the end of the labeling process,  $ug_i$  is the complement of the transitive closure of  $dg_i$ . Examples of such graphs are given in Fig.3.

##### A. First labeling step

The annotator sees the image sequences in the chronological order. For each image, he first decides if there are “precipitations” (fog included), “no precipitations”, or he may abstain. But in some exceptional cases, images of the “precipitations” class present a lower visibility when compared with any image of the “no precipitations” class. This observation allows us to fill the  $dg_i$  graph with  $n_i^p \times n_i^{np}$  new edges, where  $n_i^p$  (resp.  $n_i^{np}$ ) is the number of images with precipitations (resp. without precipitations) in the  $i - th$  sequence. Then, the annotator decides if the current image shows a lower, a higher, an “equal” or an incomparable visibility, with respect to the preceding image. The “equal” images are considered as a specific case of incomparability. The equal pairs hence lead to new edges in  $ug_i$ . But each new “equality” will also expand an equivalence relation ( $\sim_i$ ) between the images that will allow to propagate edges of  $dg_i$  and  $ug_i$ . Formally, we applied the following rule after each new annotation: **if**  $x^k \sim_i x^l$  is an “equality” between the  $k - th$  and  $l - th$  images of the sequence  $i$  **then** complete  $dg_i$  and  $ug_i$  so that for any other image  $x$  of the same sequence, one have:

$$(x^k, x) \in E_{dg_i} \Leftrightarrow (x^l, x) \in E_{dg_i}$$

$$(x, x^k) \in E_{dg_i} \Leftrightarrow (x, x^l) \in E_{dg_i}$$

$$(x^k, x) \in E_{ug_i} \Leftrightarrow (x^l, x) \in E_{ug_i}$$

where  $E_{dg_i}$  (resp.  $E_{ug_i}$ ) stands for the set of the edges of  $dg_i$  (resp.  $ug_i$ ).

##### B. Second labeling step

At the end of that first step, the following observations were made :

- new intra-sequence comparisons would improve the graph connectivity. For each new handcrafted label, transitivity closure and “equalities” will add numerous automatic labels.
- On consecutive images, incomparability was relatively rare, whereas they carry interesting information on incertitude about the ground truth. More incomparability labels were expected between non-consecutive images.

This is why the labeling process has been completed by the ordering of non-consecutive images. This ordering follows the Poset-Mergesort algorithm of [18]. This algorithm generalizes the well known mergesort

<b>Type of scene</b>	highway	field road	city street	other
	63%	18%	15%	4%
<b>Type of weather</b>	no precipitations	precipitations	doubt	thick fog
	15%	60%	25%	1%
<b>Corrupted images</b>	droplets on the lens	snowflake on the lens	filth	other
	12%	6%	2%	2%

TABLE I

DESCRIPTION OF AMOSVV. PERCENTAGES IN TABLE I ARE PROPORTIONS AMONG THE 15,727 IMAGES OF THE TRAINING SET.

	nb of sequences	nb of images	day time images	edges of $dg$	edges of $ug$
<b>Training set</b>	360	15,727	9,850	142,311	34,428
<b>Validation set</b>	66	2,234	1,435	10,997	8,141

TABLE II

RESPECTIVE SIZES OF THE TRAINING AND VALIDATION DATA SETS.  $dg$  END  $ug$  ARE DEFINED IN SECTION 4.

seq. 1	seq.2	seq.3	seq.4	seq.5
7,309	7,912	6,911	2,903	2,836

TABLE III  
SIZES OF THE TEST SEQUENCES

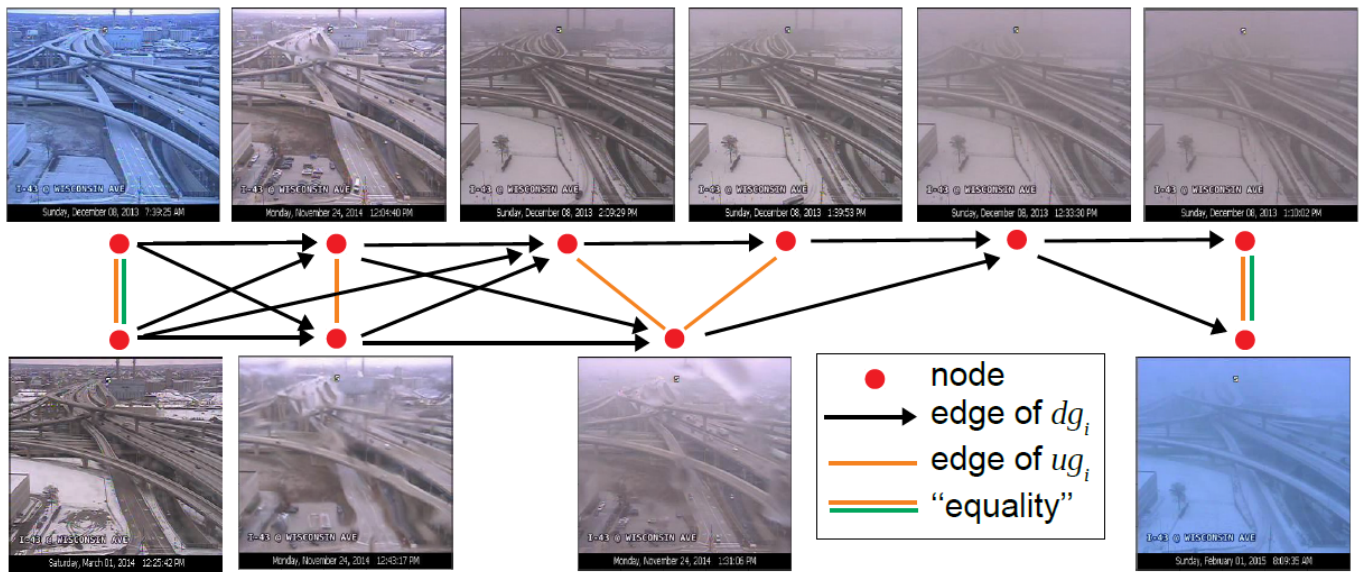


Fig. 3. Examples of manually-derived ordinal relations in the sequence  $i = 432$  (AMOS webcam 20301).

algorithm to partially ordered sets (posets). It efficiently builds a partial order from pairwise comparisons. The initial set of available comparisons corresponds to the edges of  $ug_i$  and those of the transitive closure of  $dg_i$  taken at the end of the first step.

When a comparison is not available, the human annotator takes over, and the graphs are updated.

To take the “equality” relation into account, the algorithm works on the quotient set:  $X_i/\sim_i$ , where  $X_i$  represents the set of images in sequence  $i$ . This version of the Poset-Mergesort algorithm will be available on github <sup>1</sup>.

<sup>1</sup>[https://github.com/space-latmos/poset\\_labeling](https://github.com/space-latmos/poset_labeling)

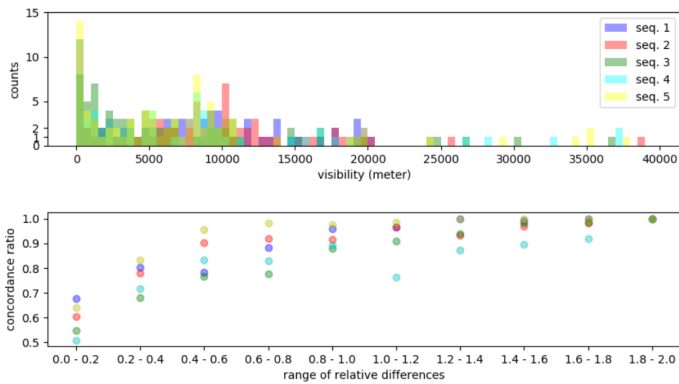


Fig. 4. Top: distribution of the visibility in the TENEBREp sequences. Bottom: Concordance ratios between manually-derived partial orders and total orders induced by instrumentally-derived visibility. Concordance ratios and relative differences of visibility are defined in the text.

In practice, the posets were built by labeling between 5 % and 10 % of all the  $n_i(n_i + 1)/2$  pairs of each sequence. This algorithm has been carried on a maximum of 40 day time image per sequence. After a restriction to subsets of daily images, the disjoint union  $dg$  (resp.  $ug$ ) of all the  $dg_i$  (resp.  $ug_i$ ) counts 153,308 edges (resp. 42,569).

### C. Comparison with the instrumentally-derived labels

This labeling method has been assessed on subsets of the TENEBREp sequences. For each sequence, 100 day time images with various visibility have been sampled and manually labeled. These five subsets will be referred to as the TENEBREp data set. We compare the posets obtained on the TENEBREp sequences with the total order induced by visibility measurements (Fig.4).

The first histogram of the Fig.4 shows the distributions of the visibility among the handcrafted subsets of TENEBREp. The global concordance ratio of the poset graph  $dg$  is the proportion of edges that are oriented in the same order as instrumentally-derived visibilities. Formally, it is defined by:  $|C|/|E_{dg}|$  where  $C = \{(i, j) \in E_{dg} \text{ s.t. } (v_i - v_j) > 0\}$ ,  $v_i$  and  $v_j$  are the instrumentally-derived visibilities corresponding to the images  $i$  and  $j$ ,  $E_{dg}$  stands for the set of the edges of  $dg$  and  $|\cdot|$  for the cardinality.

For the five TENEBREp sequences, the global concordance ratios range from 85% to 97%. But when the computation of the concordance rate is limited to image pairs of similar visibility, i.e. when the relative

difference defined by  $|x_i - x_j|/mean(x_i, x_j)$  is lower than 0.2, the concordance ratio drops (see the scatter plot of Fig.4). In section 5, these pairs are not taken into account.

## V. EXPERIMENT

Off-the-shelf classifiers were trained to predict comparisons. Inputs are formed by stacking the paired images corresponding to the  $dg_i$  and  $ug_i$  edges. The targets are defined by one of the three classes:  $>$ ,  $<$  and *incomparability* (only the first two classes in the so-called 2-classes setting).

To cope with the unbalanced size of the training sequences, the edges of the  $i$ -th sequence are selected with a frequency in proportion to  $1/n_i$ . Once a sequence is selected, the edges of  $dg_i$  are selected with a frequency of  $2/3$  and the edges of  $ug_i$  with a frequency of  $1/3$ .

Residual networks (pytorch versions of Resnet50, Resnet152, Resnext50) gave systematically lower validation scores when compared with vgg architectures. The scores presented in this section were obtained with the vgg16 architecture. The training parameters are given in Table 4. Two questions have been addressed by looking at validation scores:

- Do labeling of non-consecutive pairs improve performances? To address this question, we compared the curves of validation accuracy in two settings. In the first setting, the model is trained on the incomplete  $dg_i$ , given at the end of the first labeling step. In the second one, the model is trained on the complete  $dg_i$ . It is a 2-classes setting in both cases. The added value of the non-consecutive labels is significant, around 2-3 percentage points of accuracy.
- Could it be beneficial to take into account incomparable pairs? Again, two experimental settings were designed. In the first case, the model is trained on the complete  $dg_i$  (2-classes model). In the second case, the model is trained on the complete  $dg_i$  and  $ug_i$  (3-classes model). The validation accuracy is computed by comparison with handcrafted strict orders (the incomparability is not taken into account). When the 3-class model predicts an incomparability, we looked at the two first components of the softmax layer output to get a strict order. The second model performs slightly better (90.5% vs. 89.9%). It means that a training on

three classes does not worsen the scores on the strict order prediction task, that is the minimum requirement to select the 3-classes training.

Hereafter, we present the concordance between instrumental measurements and predictions of a 3-classes vgg16 trained on the complete graphs. Concordance ratios are computed for each sequence of the TENEBREq data set (Table 5, line 1.).  $CR_{so}$  stands for the Concordance Ratio computed on all the pairs of images that are strictly ordered by the instrument **and** by the model: pairs with a relative difference of less than 0.2 and pairs predicted incomparable are not taken into account.

For a point of comparison, we also give  $CR_{so}$  with the handcrafted pairs of the TENEBREp (line 2.). In that case, pairs with predicted or labeled incomparability are not taken into account. We observe a good agreement with all the webcams but the fourth one, with an unusual tilt angle. The concordance ratio tends to be higher with handcrafted labels.

We also want to indicate if our model excessively predicts incomparability. For that, the completeness can be used [28]. The completeness gives the excessive prediction of incomparable pairs by  $|SOP_{pred}|/|SOP_{gt}|$  where  $|SOP_{pred}|$  (resp.  $|SOP_{gt}|$ ) stands for the amount of pairs that have been strictly ordered by the model (resp. by the annotator). For the five webcams, this ratio stands in the  $[0.84, 0.94]$  interval: incomparability is not excessively predicted.

As most of the previous methods were set for intrascene prediction of a quantitative visibility, comparison is hard to make. It is yet possible to compute a Critical Success Index (CSI) for a panel of detection tasks. We take standard thresholds (250 m, 500 m, 1000 m, 1600 m, 5000 m) for the operational meteorology [21], [2]. To take a decision, we gather “pivot” images with visibility lying in the interval  $[0.9 \times t, 1.1 \times t]$  where  $t$  is a given threshold. An image of TENEBREq that is mostly predicted as strictly inferior to these pivot images is said inferior to  $t$ . The resulting CSI are, for example, favorably compared with these of Hallowell et al. [21] whereas, in this study, the training and test are made on the same set of webcams.

## VI. CONCLUSION

In this paper we presented the AMOSvv data set. It contains more than 400 road webcam sequences rich in low-visibility events. The labeling process, inspired from a mergesort algorithm, allowed an efficient manual sort of the images with respect to their apparent visibility. A simple learning to order framework has been experimented. Tests are made on independent webcams that are colocalized with weather sensors. Critical Success Indices for important detection tasks are promising. Taking into account the abstention cases gave also interesting results.

This preliminary study is to be pursued in several ways. First, night image (one half of the images) will be taken into account in an unified learning framework, following [9]. Snow cover extent and thickness will also be labeled. The dense labeling of the three parameters will allow a multi-task approach. We also will compare our method with existing ones in a near future.

## REFERENCES

- [1] World Meteorological Organization, *Guide to Meteorological Instruments and Methods of Observation No. 8*. WMO, 2014.
- [2] R. Babari, N. Hautière, É. Dumont, N. Paparoditis, and J. Misener, “Visibility monitoring using conventional roadside cameras—emerging applications,” *Transportation research part C: emerging technologies*, vol. 22, pp. 17–28, 2012.
- [3] T. M. Kwon, “Atmospheric visibility measurements using video cameras: Relative visibility,” 2004.
- [4] S. G. Narasimhan and S. K. Nayar, “Shedding light on the weather,” in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pp. 665–672, 2003.
- [5] T. Hagiwara, Y. Ota, Y. Kaneda, Y. Nagata, and K. Araki, “Method of processing closed-circuit television digital images for poor visibility identification,” *Transportation research record*, vol. 1973, no. 1, pp. 95–104, 2006.
- [6] N. Hautière, J.-P. Tarel, J. Lavenant, and D. Aubert, “Automatic fog detection and estimation of visibility distance through use of an onboard camera,” *Machine Vision and Applications*, vol. 17, no. 1, pp. 8–20, 2006.
- [7] J.-J. Liaw, S.-B. Lian, Y.-F. Huang, and R.-C. Chen, “Atmospheric visibility monitoring using digital image analysis techniques,” in *International Conference on Computer Analysis of Images and Patterns*, pp. 1204–1211, Springer, 2009.
- [8] X.-C. Yin, T.-T. He, H.-W. Hao, X. Xu, X.-Z. Cao, and Q. Li, “Learning based visibility measuring with images,” in *International Conference on Neural Information Processing*, pp. 711–718, Springer, 2011.
- [9] S. Varjo and J. Hannuksela, “Image based visibility estimation during day and night,” in *Asian Conference on Computer Vision*, pp. 277–289, Springer, 2014.
- [10] N. Graves and S. Newsam, “Using visibility cameras to estimate atmospheric light extinction,” in *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 577–584, IEEE, 2011.
- [11] N. Graves and S. Newsam, “Camera-based visibility estimation: Incorporating multiple regions and unlabeled observations,” *Ecological informatics*, vol. 23, pp. 62–68, 2014.



Model parameters	Training parameters	Best accuracies
Architecture : vgg16 [30] shape of inputs : $6 \times 448 \times 448$ initialisation : “Kaiming” [22]	batch size : 32 300 epochs optimizer : Adam [31] Initial learning rate: 0.0001 loss function: cross entropy	3-classes model : - 89.1 % (train.) - 70.6 % (valid.) Strict-order only: - 90,5 % (valid.)

TABLE IV  
LEARNING PARAMETERS.

Sequence id	1	2	3	4	5
<b>CRso</b> on TENEbREq	0.88	0.94	0.76	0.67	0.82
<b>CRso</b> on TENEbREp	0.84	0.98	0.9	0.73	0.95
<b>Completeness</b> on TENEbREp	0.93	0.86	0.84	0.85	0.94
<b>CSI</b> ( $v < 250m$ )	0.9	0.96	0.44	0.27	0.41
<b>CSI</b> ( $v < 400m$ )	0.88	0.89	0.87	0.38	0.69
<b>CSI</b> ( $v < 1000m$ )	0.96	0.93	0.83	0.1	0.63
<b>CSI</b> ( $v < 1600m$ )	0.85	0.92	0.89	0.38	0.69
<b>CSI</b> ( $v < 5000m$ )	0.81	0.78	0.5	0.5	0.71

TABLE V

LINES 1-3: CONCORDANCE RATIO OF STRICT ORDERED PAIRS (CRSO) AND COMPLETENESS ON TENEbREq AND TENEbREp.  
LINES 4-8: CRITICAL SUCCESS INDEX (CSI) FOR DIFFERENT DETECTION TASKS. THE 3-CLASSES MODEL WAS USED TO COMPUTE BOTH CRSO AND CSI.

- [12] S. Li, H. Fu, and W.-L. Lo, “Meteorological visibility evaluation on webcam weather image using deep learning features,” *Int. J. Comput. Theory Eng.*, vol. 9, no. 6, 2017.
- [13] H. Chaabani, F. Kamoun, H. Bargaoui, F. Outay, *et al.*, “A neural network approach to visibility range estimation under foggy weather conditions,” *Procedia computer science*, vol. 113, pp. 466–471, 2017.
- [14] A. Giyenko, A. Palvanov, and Y. Cho, “Application of convolutional neural networks for visibility estimation of cctv images,” in *2018 International Conference on Information Networking (ICOIN)*, pp. 875–879, IEEE, 2018.
- [15] Y. You, C. Lu, W. Wang, and C.-K. Tang, “Relative cnn-rnn: Learning relative atmospheric visibility from images,” *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 45–55, 2019.
- [16] N. Jacobs, N. Roman, and R. Pless, “Consistent temporal variations in many outdoor scenes,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, IEEE, 2007.
- [17] R. M. Rasmussen, J. Vivekanandan, J. Cole, B. Myers, and C. Masters, “The estimation of snowfall rate using visibility,” *Journal of Applied Meteorology*, vol. 38, no. 10, pp. 1542–1563, 1999.
- [18] C. Daskalakis, R. M. Karp, E. Mossel, S. J. Riesenfeld, and E. Verbin, “Sorting and selection in posets,” *SIAM Journal on Computing*, vol. 40, no. 3, pp. 597–622, 2011.
- [19] C.-H. Luo, C.-Y. Wen, C.-S. Yuan, J.-J. Liaw, C.-C. Lo, and S.-H. Chiu, “Investigation of urban atmospheric visibility by high-frequency extraction: Model development and field test,” *Atmospheric Environment*, vol. 39, no. 14, pp. 2545–2552, 2005.
- [20] L. Xie, A. Chiu, and S. Newsam, “Estimating atmospheric visibility using general-purpose cameras,” in *International Symposium on Visual Computing*, pp. 356–367, Springer, 2008.
- [21] R. Hallowell, M. Matthews, and P. Pisano, “An automated visibility detection algorithm utilizing camera imagery,” in *23rd Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology (IIPS)*, 2007.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [23] N. Hautière, R. Babari, É. Dumont, R. Brémond, and N. Paparoditis, “Estimating meteorological visibility using cameras: A probabilistic model-driven approach,” in *Asian Conference on Computer Vision*, pp. 243–254, Springer, 2010.
- [24] D. Parikh and K. Grauman, “Relative attributes,” in *2011 International Conference on Computer Vision*, pp. 503–510, IEEE, 2011.
- [25] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133–142, ACM, 2002.
- [26] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman, “Learning ordinal relationships for mid-level vision,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 388–396, 2015.
- [27] S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4353–4361, 2015.
- [28] W. Cheng, M. Rademaker, B. De Baets, and E. Hüllermeier, “Predicting partial orders: ranking with abstention,” in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 215–230, Springer, 2010.
- [29] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, G. De Chiara, P. Dahlgren, D. Dee, M. Diamantakis, R. Dragani, J. Flem-

- ming, R. Forbes, M. Fuentes, A. Geer, L. Haimberger, S. Healy, R. J. Hogan, E. Hólm, M. Janisková, S. Keeley, P. Laloyaux, P. Lopez, C. Lupu, G. Radnoti, P. de Rosnay, I. Rozum, F. Vamborg, S. Villaume, and J.-N. Thépaut, “The era5 global reanalysis,” *Quarterly Journal of the Royal Meteorological Society*, vol. n/a, no. n/a.
- [30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [31] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in *Advances in neural information processing systems*, pp. 3581–3589, 2014.