1 *Article*

Deep convolutional neural networks capabilities for binary classification of polar mesocyclones in satellite mosaics

5 **Mikhail Krinitskiy** [1,*]**, Polina Verezemskaya** [1,2]**, Kirill Grashchenkov** [1,3]**, Natalia Tilinina** [1]**,
6 Sergey Gulev** [1] **and Matthew Lazzara** [4]

7 [1] Shirshov Institute of Oceanology, Russian Academy of Sciences, Moscow, Russia; info@ocean.ru
8 [2] Research Computing Center of Lomonosov Moscow State University, Moscow, Russia
9 [3] Moscow Institute of Physics and Technology, Moscow, Russia
10 [4] University of Wisconsin-Madison and Madison Area Technical College, Madison, Wisconsin, USA
11 ***** Correspondence: krinitsky@sail.msk.ru; Tel.: +7-926-141-6200

13 **Abstract:** Polar mesocyclones (MCs) are small marine atmospheric vortices. The class of intense
14 MCs, called polar lows, are accompanied by extremely strong surface winds and heat fluxes and
15 thus largely influencing deep ocean water formation in the polar regions. Accurate detection of
16 polar mesocyclones in high-resolution satellite data, while challenging, is a time-consuming task,
17 when performed manually. Existing algorithms for the automatic detection of polar mesocyclones
18 are based on the conventional analysis of patterns of cloudiness and involve different empirically
19 defined thresholds of geophysical variables. As a result, various detection methods typically reveal
20 very different results when applied to a single dataset. We develop a conceptually novel approach
21 for the detection of MCs based on the use of deep convolutional neural networks (DCNNs). As a
22 first step, we demonstrate that DCNN model is capable of performing binary classification of
23 500x500km patches of satellite images regarding MC patterns presence in it. The training dataset is
24 based on the reference database of MCs manually tracked in the Southern Hemisphere from satellite
25 mosaics. We use a subset of this database with MC diameters falling in the range of 200-400 km.
26 This dataset is further used for testing several different DCNN setups, specifically, DCNN built
27 "from scratch", DCNN based on VGG16 pre-trained weights also engaging the Transfer Learning
28 technique, and DCNN based on VGG16 with Fine Tuning technique. Each of these networks is
29 further applied to both infrared (IR) and a combination of infrared and water vapor (IR+WV)
30 satellite imagery. The best skills (97% in terms of the binary classification accuracy score) is achieved
31 with the model that averages the estimates of the ensemble of different DCNNs. The algorithm can
32 be further extended to the automatic identification and tracking numerical scheme and applied to
33 other atmospheric phenomena characterized by a distinct signature in satellite imagery.

34 **Keywords:** deep learning, convolutional neural networks, polar mesocyclones, satellite data
35 processing, pattern recognition

36

37 **Nomenclature**

38   BCE – binary cross-entropy
39   CNN – convolutional neural network
40   DA – dataset augmentation technique
41   DCNN – deep convolutional neural network
42   DL – deep learning
43   Do – Dropout technique
44   FC – fully-connected

45   FCNN – fully-connected neural network
46   FT – Fine Tuning
47   FNR – false negative rate
48   FPR – false positive rate
49   IR – infrared
50   MC – mesocyclone
51   NH – Northern Hemisphere
52   PL – polar low
53   ROC – receiver operator characteristic
54   AUC ROC – area under the curve of receiver operator characteristic
55   SH – Southern Hemisphere
56   SOMC – Shirshov Institute of Oceanology mesocyclone dataset for Southern Ocean
57   TL – Transfer Learning
58   TNR – true negative rate
59   TPR – true positive rate
60   VGG16 – the DCNN proposed by Visual Geometry Group (University of Oxford) [1]
61   WV – water vapor

1. Introduction

63   Polar mesoscale cyclones (MCs) are high-latitude marine atmospheric vortices. Their sizes range
64 from 200 to 1000 km with lifetimes typically spanning from 6 to 36 hours [2]. A specific intense type
65 of mesocyclones, the so-called polar lows (PLs) is characterized by surface winds of more than 15 m/s
66 and strong surface fluxes. These PLs have a significant impact on local weather conditions causing
67 rough seas. Being relatively small in size (compared to the extratropical cyclones), PLs contribute
68 significantly to the generation of extreme air-sea fluxes and initialize intense surface transformation
69 of water masses resulting in the formation of ocean deep water [3–5]. These processes are most intense
70 in the Weddel and Bellingshausen Seas in the Southern Hemisphere (SH) and in the Labrador,
71 Greenland and Irminger Seas in the Northern Hemisphere (NH).
72   One potential source of data is reanalyses. However, MCs, being critically important for many
73 oceanographic and meteorological applications, are only partially detectable in different reanalysis
74 datasets, primarily due to the inadequate resolution. Studies [4,6–9] have demonstrated the
75 significant underestimation of both number of mesocyclones and wind speeds by modern reanalyses
76 in contrast with satellite observations of MCs cloud signatures and wind speeds. This hints that the
77 spatial resolution of modern reanalyses is still not good enough for reliable and accurate detection of
78 MCs. Press et al. argued for at least 10 x 10 grid points is necessary for effective capturing the MC [10].
79 This implies a 30 km spatial resolution in the model or reanalysis is needed for detecting MC with
80 the diameter of 300 km. Some studies [6,11] have demonstrated that 80% (64%) of MCs (PLs) in the
81 SH (NH) are characterized by the diameters ranging from 200 to 500 km (250 to 450 km for NH in
82 [11]). The most recent study of Smirnova and Golubkin [12] revealed that only 70% of those could be
83 sustainably represented even in the very high-resolution Arctic System Reanalysis (ASR) [13]. At the
84 same time only 53% of the observed MCs characterized by diameters less than 200 km [6] are
85 sustainably represented in ASR [12]. It was also shown [4,6,7] that both number of MCs and
86 associated winds in modern reanalyses are significantly underestimated compared to satellite
87 observations of cloud signatures of MCs and satellite scatterometer observations of MC winds.
88   One might argue for the use of operational analyses for detecting MCs. However, these products
89 are influenced by the changes in the numerics of a model and physics parameterization schemes with
90 newly developed ones, and by the changes of the performance of data assimilation system and the
91 amount of assimilated data. This leads to artificial trends at climatological timescales. In several
92 studies, automated cyclone tracking algorithms originally developed for mid-latitude cyclones were
93 adapted for MCs identification and tracking [14–16]. These algorithms were applied to the
94 preprocessed (spatially filtered) reanalysis data and delivered climatological assessments of MCs

95   activity in reanalyses or revealed the direction for their improvement. However, reported estimates
96   of MCs numbers, sizes and lifecycle characteristics vary significantly in these studies.
97   Zappa et al. [14] shows that ECMWF operational analysis makes it possible to detect up to 70%
98   of the observed PLs, which is higher than ERA40 and ERA-Interim reanalyses (24%, 45% or 55%
99   depending on the procedure of tracking and the choice of reanalysis [7,14]). One bandpass filter in
100   conjunction with different combinations of criteria used for the post-processing of the MC tracking
101   results may result in a 30% spread in the number of PLs [14]. Observational satellite-based
102   climatologies of MCs and PLs [6,11,17–20] consistently reveal a mean vortex diameter of 300-350 km.
103   In a number of reanalysis-based automated studies [15,21], the upper limit of MC and PL diameters
104   was set to 1000 km, resulting in the mean values between 500 and 800 km. Thus, the estimates of MC
105   sizes are still inconsistently derived with automated tracking algorithms. This inconsistency contrasts
106   with the estimates for midlatitude cyclones' characteristics derived with the ensemble of tracking
107   schemes [22] applied to a single dataset.
108   Satellite imagery of cloudiness is another data source for identification and tracking of MCs.
109   These data allow for visual identification of cloud signatures associated with MCs. However, the
110   manual procedure re Carleton quires enormous effort to build a long enough dataset. Pioneering work
111   of Wilhelmsen [23] used ten years of consecutive synoptic weather maps, coastal observational
112   stations and several satellite images over the Norwegian and Barents Seas to describe local PLs
113   activity. Later in the 1990s, the number of instruments and satellite crossovers increased. It provoked
114   many studies [17,24–29] evaluating characteristics of MCs occurrence and lifecycle in different
115   regions of both NH and SH. These studies identified major MCs generation regions, their dominant
116   migration directions, and cloudiness signature types associated with MCs. Increases in the amount
117   of satellite observations allowed for the development of robust regional climatologies of MCs
118   occurrence and characteristics. For the SH, Carleton [28] used twice daily cloudiness imagery of West
119   Antarctica and classified for the first time four types of cloud signatures associated with PLs (comma,
120   spiral, transitional type, and merry-go-round). This classification has been confirmed later in many
121   works and is widely used now. Harold et al. [17,27] used daily satellite imagery for building one of
122   the most detailed datasets of MC characteristics for the Nordic Seas (Greenland, Norwegian, Iceland
123   and Northern Seas). Also, Harold et al. [17,27] developed a detailed description of the conventional
124   methodology for the identification and tracking of MCs using satellite IR imageries.
125   There are also several studies regarding polar MCs and PLs activity in the Sea of Japan.
126   Gang et al. [30] conducted the first long-term (three winter months) research of PLs in the Sea of Japan
127   based on visible and IR imagery from the geostationary satellite with hourly resolution. In the era of
128   multi-sensor satellite observations, Gurvich and Pichugin [31] developed the 9-year climatology of
129   polar MCs based on water vapor, cloud water content and surface wind satellite data over the
130   Western Pacific. This study reveals a mean MCs diameter of 200-400 km as well.
131   As these examples illustrate, most studies of MCs activity are regional [11,18,19,32,33] and cover
132   relatively short time periods [6] due to the very costly and time-consuming procedure of visual
133   identification and tracking of MCs. Thus, development of the reliable long-term (multiyear) dataset
134   covering the whole circumpolar Arctic or Antarctic remains a challenge.
135   Recently, machine learning methods have been found to be quite effective for the classification
136   of different cloud characteristics such as solar disk state and cloud types. There are studies in which
137   different machine learning techniques are used for recognizing cloud types [34–36]. Methodologies
138   employed include deep convolutional neural networks (DCNNs [37,38]), k-nearest-neighbor
139   classifier (KNN) and Support Vector Machine (SVM) and fully-connected neural networks (FCNNs).
140   Krinitskiy [39] used FCNNs for the detection of solar disk state and reported very high accuracy
141   (96.4%) of the proposed method. Liu et al. [40] applied DCNNs to the fixed-size multichannel images
142   to detect extreme weather events and reported the success score of the detection of 89 to 99%. Huang
143   et al. [41] applied the neural network "DeepEddy" to the synthetic aperture radar images for
144   detection of ocean meso- and submesoscale eddies. Their results are also characterized by high
145   accuracy exceeding 96% success rate. However, Deep Learning (DL) methods have never been
146   applied for detecting MCs.

DCNNs are known to demonstrate high skills in classification, pattern recognition, and semantic segmentation, when applied to 2-dimensional (2D) fields, such as images. The major advantage of DCNNs is the depth of processing of the input 2D field. Similarly to the processing levels of satellite data (L0, L1, L2, L3, etc.), which allow retrieving, e.g. wind speed (L2 processing) from the raw remote measurements (L0), DCNNs are dealing with multiple levels of subsequent non-linear processing of an input image. In contrast to the expert-designed algorithms, the neural network levels of processing (so-called layers) are built in a manner that is common within each specific layer type (convolutional, fully-connected, subsampling, etc.). During the network training process, these layers of a DCNN acquire the ability to extract a broad set of patterns of different scales from the initial data [42–45]. In this sense, a trained DCNN closely simulates the visual pattern recognition process naturally used by a human operator. There exist several state-of-the-art network architectures such as "AlexNet" [37], "VGG16" and "VGG19" [1], "Inception" of several subversions [46], "Xception" [47] and residual networks [48]. Each of these networks has been trained and tested using a range of datasets including the one that is considered as a "reference" for the further image processing, the so-called ImageNet [49]. Continuous development of all DCNNs aims to improve the accuracy of the ImageNet classification. Today, the existing architectures demonstrate high accuracy with the error rate from 2% to 16% [50].

A DCNN by design closely simulates the visual recognition process. IR and WV satellite mosaics can be interpreted as images. Thus, assuming that a human expert detects MCs on these mosaics on the basis of his visual perception, application of DCNN appears to be a promising approach to this problem. Liu et al. [40] described a DCNN applied to the detection of tropical cyclones and atmospheric rivers in the 2D fields of surface pressure, temperature and precipitation stacked together into "image patches." However, the proposed approach cannot be directly applied to the MC detection. This method is skillful for the detection of large-scale weather extremes that are discernible in reanalysis products. However, as noted above, MCs have poorly observable footprint in geophysical variables of reanalyses.

In this study, we apply the DL technique [51–53] to the satellite IR and WV mosaics distributed by Antarctic Meteorological Research Center [54,55]. This allows for the automated recognition of MCs cloud signatures. Our focus here is exclusively on the capability of DCNNs to perform a binary classification task regarding MCs patterns presence in patches of satellite imagery of cloudiness and/or water vapor, rather than on the DCNN-based MC tracking. This will indicate that a DCNN is capable of learning the hidden representation that is in accordance with the data and the MCs detection problem.

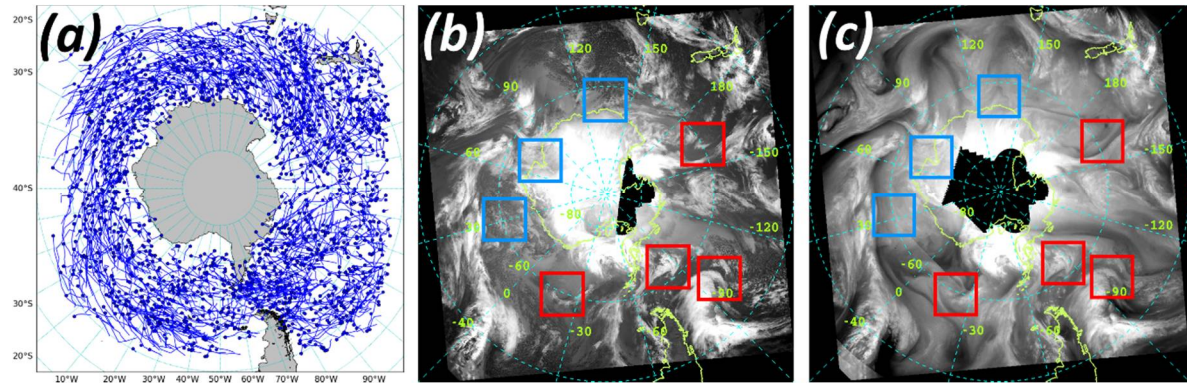
The paper is organized as follows. Section 2 describes the source data based on MC trajectories database [6]. Section 3 describes the development of the MC detection method based on deep convolutional neural networks and necessary data preprocessing. In Section 4 we present the results of the application of the developed methodology. Section 5 summarizes the paper with the conclusions and provides an outlook.

2. Data

For the training of DCNNs, we use MCs dataset for the Southern Ocean (SOMC, http://sail.ocean.ru/antarctica/) consisting of 1735 MC trajectories, resulting in 9252 MC locations and associated estimates of MC sizes [6] for the 4-months period (June, July, August, September) of 2004 (Figure 1a). The dataset was developed by visual identification and tracking of MCs using 976 consecutive 3-hourly satellite IR (10.3 - 11.3 micron) and WV (~6.7 microns) mosaics provided by the Antarctic Meteorological Research Center (AMRC) Antarctic Satellite Composite Imagery (AMRC ASCI) [54,55]. These mosaics are available online (https://amrc.ssec.wisc.edu/data/) and are composites of geostationary and polar-orbiting satellite observations (GOES East and West, Meteosat, MTSAT, NOAA satellites, METOP, FY-2, Aqua, Terra etc.). This mosaics dataset is maintained by the AMRC [55]. The SOMC dataset contains longitudes and latitudes of MC centers at each 3-hourly time step of the MC track as well as MC diameter and the cloudiness signature type

197 through the MC life cycle [6]. These characteristics were used along with the associated cloudiness
198 patterns of MCs from the initial IR and WV mosaics for training DCNNs.

199   AMRC ASCI mosaics spatially combine observations from geostationary and polar-orbiting
200 satellites and cover the area to the South of ~40°S with 3-hourly temporal and 5 km spatial resolution
201 (Fig. 1bc). While the IR channel is widely used for MCs identification [17,18,27,28,33], we also
202 additionally employ the WV channel imagery which provides a better accuracy over the ice-covered
203 ocean, where the IR images are potentially incorrect.

204

206 **Figure 1.** The input for the deep convolutional neural networks (DCNNs). (a) Trajectories of all
207   mesocyclones (MCs) in Southern Ocean MesoCylones (SOMC) dataset, blue dots mark the point of
208   generation of MC. Snapshots of satellite mosaics for SH for (b) InfraRed (IR) and (c) Water Vapor
209   (WV) channels at 00:00 UTC 02/06/2004. The red/blue squares indicate patches centered over the MCs
210   (red squares) and those having no MC cloudiness signature in (blue) being cut from the mosaics for
211   DCNNs training.

212 **3. Methodology**

213 *3.1. Data preprocessing*

214   For training models, we first co-located a square (patch) of 100x100 mosaic pixels (500x500 km)
215 with each MC center location from SOMC dataset (9252 locations in total) (Figure 2a-d). Since the
216 distance between MCs in the multiple systems such as the merry-go-round pattern may be
217 comparable to each MC diameter, and to ensure that (i) each patch covers only one MC and (ii) covers
218 it completely, we require that MC diameters fall into 200-400 km range. Hereafter we call this set of
219 samples 'the true samples'. The chosen set of true samples includes 67% of the whole population of
220 samples in SOMC dataset.

221   We additionally built the set of 'false samples' for DCNNs training. False samples were
222 generated from the patches that do not consist of MC-associated cloudiness signatures (Figure 2e-h)
223 according to the SOMC dataset. Table 1 summarizes the numbers of true and false samples that both
224 make up the source dataset for our further analysis of IR and WV mosaics. The total number of
225 snapshots used (both IR and WV) is 11189. The true samples are 6177 (55%) of them, and 5012 (45%)
226 are the false samples (see Fig. 2). In order to unify images in the dataset, we normalized them by the
227 maximum and the minimum brightness temperature (in the case of IR) over the whole dataset:

228

$$x_{norm} = \frac{x - \min(X)}{\max(X) - \min(X)}, \qquad (1)$$

229

230 where x denotes the individual sample (represented by a matrix of 100x100 pixels), X is the whole
231 dataset of 11189 IR snapshots. The same normalization was applied to WV snapshots.

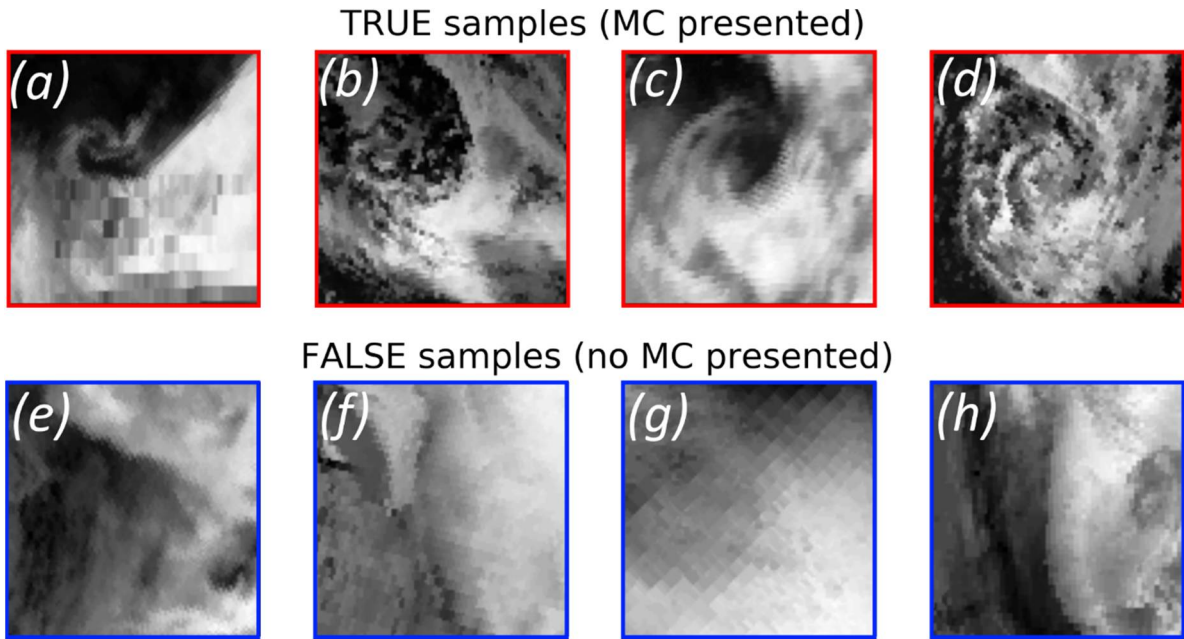



Figure 2. Examples (IR only) of true and false samples for DCNNs training and testing of DCNNs results assessment. 100x100 grid points (500x500km) patches of IR mosaics for (a-d) true samples and false (e-h) samples.

3.2. Formulation of the problem

We consider MC identification as a binary classification problem. We use the set of true and false samples (Figure 2) as input ("objects" herein). We have developed two DCNN architectures following two conditional requirements: either (i) the object is described by the IR image only or (ii) the object is described by both IR and WV images. Since the training dataset is almost target-balanced (see Table 1), assuming ~50/50 ratio of true/false samples, we further use the accuracy score as the measure of the classification quality. The accuracy score cannot be used as a reliable quality measure of any machine learning method in the case of the unbalanced training dataset. For example, in the case of a highly unbalanced dataset with the true/false ratio being 95/5 it is easy to achieve 95% accuracy score by just forcing the model to produce only the true outcome. Thus, balancing the source dataset with false samples is critical for building the reliable classification model.

Table 1. Total number of true and false samples.

	True samples	False samples	Total samples
IR	6177 (55%)	5012 (45%)	11189 (100%)
WV	6177 (55%)	5012 (45%)	11189 (100%)

3.3. Justification of using DCNN

There is a set of best practices commonly used to construct DCNNs for solving classification problems [56]. While building and training DCNNs for MCs identifications, we applied the technique proposed by LeCun [42]. This technique implies the usage of consecutive convolutional layers which detect spatial data patterns, alternating with subsampling layers which reduce the sample dimensions. The set of these layers is followed by a set of so-called fully-connected (FC) layers representing a neural classifier. The whole model built in this manner represents a non-linear classifier capable of directly predicting a target value for the input sample. A very detailed description of this model architecture can be found in [42]. We will further term the FC layers set as "FC classifier," and the preceding part containing convolutional and pooling layers as "convolutional core" (see Figures 3,4). The outcome of the whole model is the probability of MC presence in the input sample.

260   While handling multiple concurrent and spatially aligned geophysical fields, it is important to
261 choose a suitable approach. LeCun [42] proposed the DCNN focused on the processing of only
262 grayscale images – meaning just one 2D field. In order to handle multiple 2D fields, they may be
263 stacked together to form a 3D matrix by analogy with colorful images which have three color
264 channels: red, green and blue. This approach can be applied when one uses pre-trained networks like
265 AlexNet [37], VGG16[1], ResNet [48] or similar architectures because of the original purpose of these
266 networks to classify colorful images. However, this approach should be exploited carefully when
267 applied to geophysical fields, because the mentioned networks were trained using massive datasets
268 (e.g., ImageNet) of real photographed scenes, which means specific dependencies laying between
269 channels (red, green and blue) within each image. In contrast to the stacking approach applied by
270 Liu et al. [40], we use separate CNN branch for each channel (IR and WV) to ensure that we are not
271 limiting the overall quality of the whole network (see Fig. 4). In the following, we describe in details
272 each DCNN architecture for both cases: IR+WV (Fig. 4) and IR alone (Fig. 3).

273   Since we consider the binary classification, and the source dataset is almost target-balanced
274 (see Tab. 1), we use as a quality measure the accuracy score or Acc which is a rate of objects, classified
275 correctly compared to the ground truth:

$$Acc = \frac{1}{\|\mathcal{T}\|} \sum_{\mathcal{T}} [\hat{y}_i = y_i] \,, \tag{2}$$

276 where $\mathcal{T}$ denotes the dataset and $\|\mathcal{T}\|$ is its total samples count; y_i is expert-defined target value
277 (ground truth), $\hat{y}_i$ is the model decision whether the i-th object contain MC.
278 In addition to the baseline which is the network proposed in [42], we applied a set of additional
279 approaches commonly used to improve the DCNN accuracy and generalization ability
280 (see Appendix A). Specifically, we used Transfer Learning (TL) [57–62] with the VGG16 [1] network
281 pre-trained on ImageNet [49] dataset; Fine Tuning (FT) [63], Dropout (Do) [64] and dataset
282 augmentation (DA) [65] (see Appendix A). With these techniques applied in various combinations,
283 we constructed six DCNN architectures that are summarized in Table 2. All of these architectures are
284 built in a common manner: the FC classifier follows the one- (for IR only) or two-branched (for
285 IR+WV) convolutional core. If the convolutional core is one-branched, its output itself is input data
286 for the corresponding FC classifier. If the convolutional core is two-branched, the concatenation
287 product of their outputs is the input data for the corresponding FC classifier. The very detailed
288 description of the constructed architectures is presented in Appendix A. For each DCNN structure
289 we trained a set of models as described in detail in section 3.5. We also applied ensemble averaging
290 (see Appendix A) of a set of models of identical configuration via averaging probabilities of true class
291 for each object of the dataset. We term these six ensemble-averaged models the "second-order"
292 models. We also applied ensemble averaging per sample of all trained DCNNs trained in this work.
293 We term this model the "third-order" model. Each of these models was trained using the method of
294 backpropagation of error (BCE loss, see Appendix A) [66] denoted as "backprop training" in Figures
295 3 and 4.

296 *3.4. Proposed DCNN architectures*

297   Six DCNNs that we have constructed are able to perform binary classification on satellite
298 mosaics data (IR alone or IR+WV) represented as grayscale 100 x 100 pixels images:
299 1.   CNN #1. This model is built "from scratch" which means we have not used any pre-trained
300   networks. CNN #1 is built in the manner proposed in [36]. We varied sizes of convolutional
301   kernels of each convolutional layers from 3x3 to 5x5. We also varied sizes of subsampling layers'
302   receptive fields from 2x2 to 3x3. For each convolutional layers, we varied the number of
303   convolutional kernels: 8, 16, 32, 64 and 100. The network convolutional core consists of three
304   convolutional layers alternated with subsampling layers. Each pair of convolutional and
305   subsampling layers is followed by a dropout layer. CNN #1 is one-branched, and objects are
306   described by IR 500x500 km satellite snapshots only.

2. CNN #2. This model is built "from scratch" with two separate branches - for IR and WV data. The convolutional core of each branch is built in the same manner as the convolutional core for CNN #1 and as proposed in [42]. We varied the same parameters of the structure here in the same ranges as for CNN #1.

3. CNN #3. This model is built with TL approach. We used VGG16 pre-trained convolutional core to construct this model. None of VGG16 weights were optimized within this model, and only the weights of the FC classifier were trainable. This model is one-branched, and objects are described by IR 500x500 km satellite snapshots only. CNN #3 structure is shown in Fig. 3.

4. CNN #4. This model is two-branched, and each branch of its convolutional core is built with TL approach, in the same manner as the convolutional core of CNN #3. Input data are IR and WV. None of VGG16 weights of this model in any of the two branches were optimized, and only the weights of the FC classifier were trainable. CNN #4 structure is shown in Fig. 4.

5. CNN #5 is built with both TL and FT approaches. We built the convolutional core of this model with the use of VGG16 pre-trained network. VGG16 convolutional core consists of five similar blocks of layers. For the CNN #5 we turned the last of these five blocks to be trainable. This model is one-branched, and objects are IR 500x500 km satellite snapshots only. CNN #5 structure is shown in Fig. 3.

6. CNN #6 is two-branched, and branches of its convolutional core are built in the same manner as the convolutional core of CNN #5. For the CNN #6 we turned the last of five blocks of each VGG16 convolutional cores to be trainable. Input data are IR and WV 500x500 km satellite snapshots of dataset samples. CNN #6 structure is shown in Fig. 4.

3.5. Computational experiment design

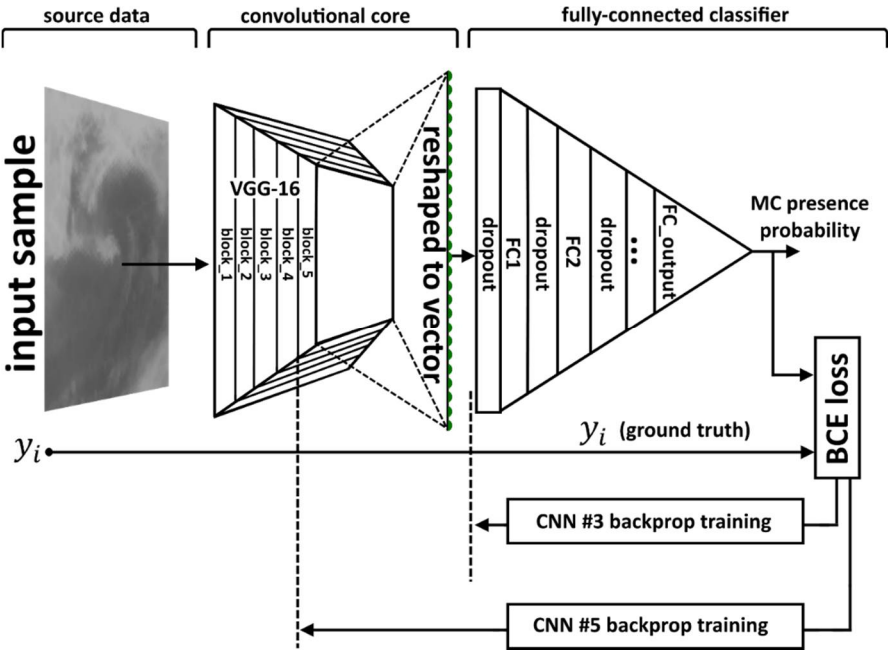
The following hyper-parameters are included in each of the six networks:

- Size (number of nodes) of the first layer of FC classifier (denoted as FC1 in Figures 3,4)
- Convolutional kernels count for each convolutional layer (only applies to CNN #1 and CNN #2)
- Sizes of convolutional kernels (only applies to CNN #1 and CNN #2)
- Sizes of receptive fields of subsampling layers (only applies to CNN #1 and CNN #2)

The whole dataset was split into training (8952 samples) and testing (2237 samples) sets stratified by target value meaning that each set has the same (55:45) ratio of true/false samples as the whole dataset (i.e., 4924:4028 and 1253:984 samples in training and testing sets correspondingly). We have conducted hyper-parameters optimization for each of these DCNNs using stratified K-fold (K=5) cross-validation approach. After this optimization, we trained several (typically 14-18) models with the best hyper-parameters configuration on the training set for each architecture (architecture-specific models). Then we excluded models with the maximal and minimal accuracy score estimated with the cross-validation approach, from this set of architecture-specific models for each of the six architectures. The remaining architecture-specific models were evaluated on the testing set, which was never seen by these models. We estimated the accuracy score for each individual model and the variance of accuracy score for the particular architecture with the best hyper-parameters combination (see Table 2).

With the ensemble averaging approach, we evaluated the second-order models on the "never-seen by the model" testing set. As described in section 3.3 we estimated the optimal probability threshold p_{th} for each second-order model and for the third-order model (see Table 2) for the best accuracy score estimation. These scores are treated as the quality measure of each particular architecture.

Numerical optimization and evaluation of models were performed at the Data Center of Far Eastern Branch of Russian Academy of Sciences (FEB RAS) [67] and DL computational resources of Sea-Air Interactions Laboratory of Shirshov Institute of Oceanology of Russian Academy of Sciences (IORAS, https://sail.ocean.ru/). Exploited computational nodes contain two graphics processing units (GPU) NVIDIA Tesla P100 16GB RAM. With these resources, the total GPU time of calculations is 3792 hours.

358

359 **Figure 3.** CNN #3 and CNN #5 structures. The dots along the "reshaped to vector" line denote
360 elements of the convolutional core output reshaped to a vector, which is the fully-connected classifier
361 input data.

362 **Figure 4.** CNN #4 and CNN #6 structures. The dots along the "reshaped to vector" and "concatenated
363 features vector" lines denote elements of convolutional cores outputs reshaped to vectors, which are,
364 being concatenated to a combined features vector, the fully-connected classifier input data.

4. Results

The designed DCNNs were applied to detect of Antarctic MCs for the period from June to September 2004. Summary of the results of the application of six models is presented in Table 2. As we noted above, each model is characterized by the utilized data source (IR alone or IR+WV, columns "IR" and "WV" in Table 2). These DCNNs are further categorized according to a chosen set of applied techniques in addition to the basic approach (see Table 2 legend). Table 2 also provides accuracy scores and probability thresholds estimated as described in section 3.5, for the individual, second- and third-order models of each architecture.

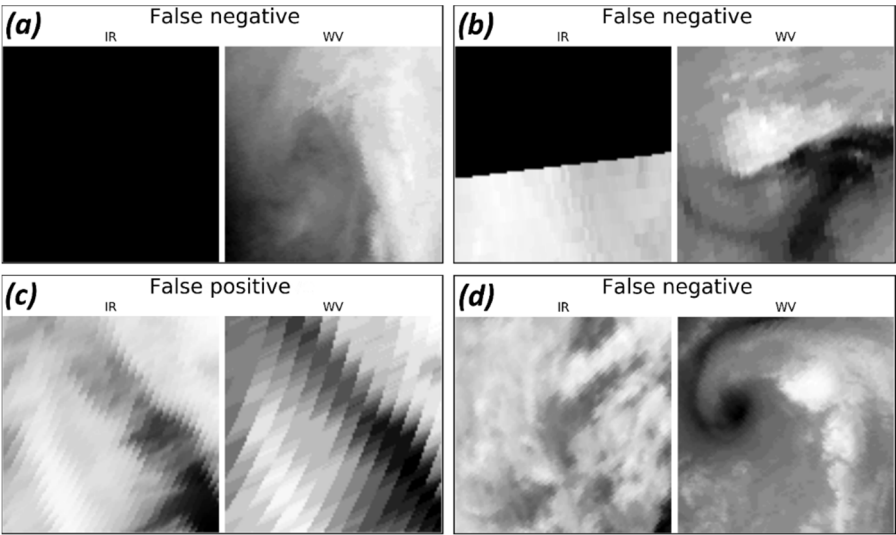
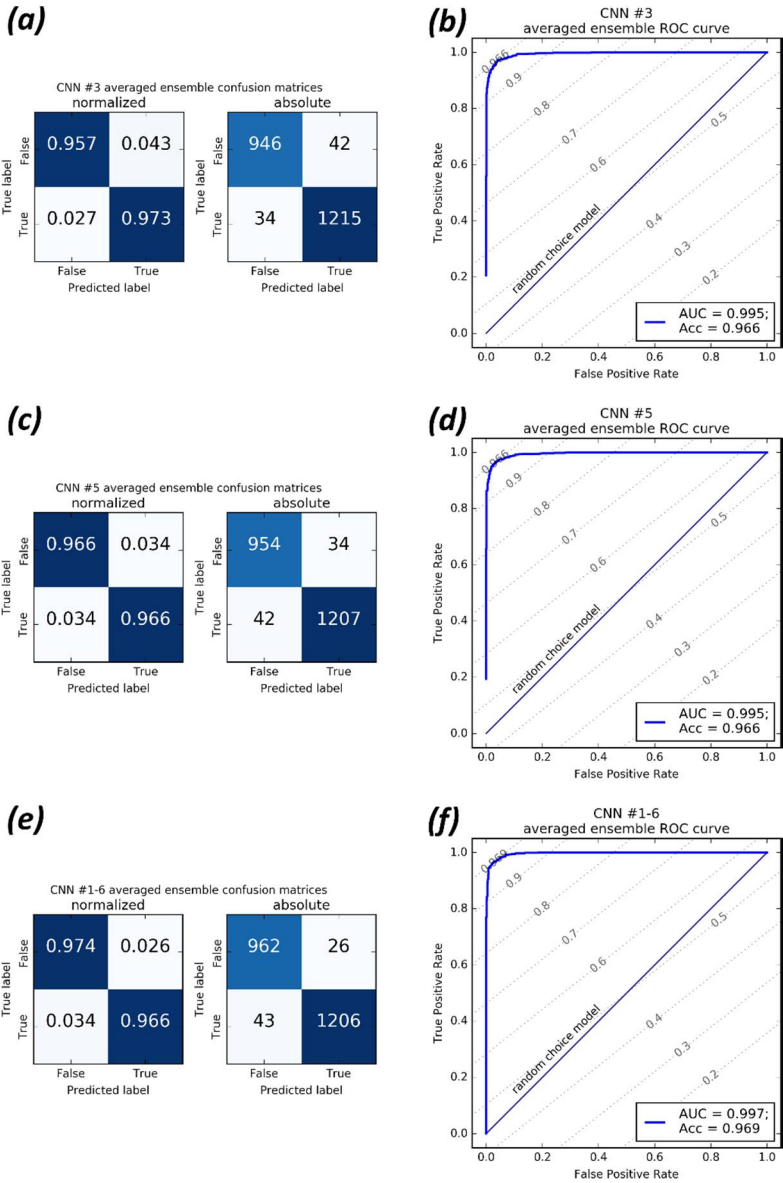
Table 2. Accuracy score of each model with the best hyper-parameters combination. BA - basic approach [42], TL - Transfer Learning, FT - Fine Tuning, Do - dropout, DA - dataset augmentation. *Acc* is the accuracy score averaged across models of the particular architecture. AsEA is the accuracy score of the ensemble averaged models with the optimal probability threshold. p_{th} is the optimal probability threshold value.

model name	IR	WV	BA	TL	FT	Do	DA	*Acc*	AsEA	p_{th}
CNN #1	X	-	X	-	-	X	X	86.89 ± 1.1 %	89.3 %	0.381
CNN #2	X	X	X	-	-	X	X	94.1 ± 1.4 %	96.3 %	0.272
CNN #3	X	-	X	X	-	X	X	95.8 ± 0.1 %	96.6 %	0.556
CNN #4	X	X	X	X	-	X	X	95.5 ± 0.3 %	96.3 %	0.526
CNN #5	X	-	X	X	X	X	X	96 ± 0.2 %	96.6 %	0.5715
CNN #6	X	X	X	X	X	X	X	95.7 ± 0.2 %	96.4 %	0.656
Third-order model CNN #1-6 averaged ensemble									97%	0.598

As shown in Table 2, CNN #3 and CNN #5 demonstrated the best accuracy among the second-order models on a never-seen subset of objects. The best combination of hyper-parameters for these networks is presented in Appendix B. Confusion matrices and receiver operating characteristic (ROC) curves for these models are shown in Fig. 6 a-d. Confusion matrices, and ROC curves for all evaluated models are presented in Appendix C. Figure 6 clearly confirms that these two models perform almost equally for the true and the false samples. According to Table 2, the best accuracy score is reached using different probability thresholds for each second- or third-order model.

Comparison of CNN #1, CNN #2, on the one hand, and the remaining models, on the other hand, shows that DCNNs built with the use of TL technique demonstrate better performance compared to the models built "from scratch". Moreover, the accuracy score variances of CNN #1 and CNN #2 are higher than for the other architectures. Thus, models built with TL approach seem to be more stable, and their generalization ability is better, compared to models built "from scratch."

Comparing CNN #1 and CNN #2 qualities, we may conclude that the use of an additional data source (WV) results in the significant increase of the model accuracy score. Comparison of models within each pair of the network configurations (CNN #3 vs. CNN #5; CNN #4 vs. CNN #6) demonstrates that FT approach does not provide significant improvement of the accuracy score in case of such a small size of the dataset. It is also obvious that the averaging over the ensemble members does increase the accuracy score from 0.6% for CNN #5 to 2.41% for CNN #1. However, in some cases, these score increases are comparable to the corresponding accuracy standard deviations.

It is also clear from the last row of Table 2, that the third-order model, which averages probabilities estimated by all trained models CNN #1-6, produces the accuracy of $Acc = 97\%$ which outperforms all scores of individual models and second-order ensemble models. ROC curve and confusion matrices for this model are presented in Figure 6ef.

405
406   **Figure 5.** False classified objects.

407   **Figure 6.** Confusion matrices and receiver operating characteristic curve for (a,b) CNN #3 and (c,d)
408   CNN #5, both with the ensemble averaging approach applied (second-order models); and (e,f) third-
409   order model CNN #1-6 averaged ensemble.

410   Figure 5 demonstrates four main types of false classified objects. The first and the second types
411   are the ones for which IR data are missing completely or partially. The third type is the one for which
412   the source satellite data were suspected to be corrupted. These three types of classifier errors
413   originating from the lack of source data or the corruption of source data. For the fourth type, the
414   source satellite data were realistic but the classifier has made a mistake. Thus, some of false
415   classifications are model mistakes, and some are associated with the labeling issue where human
416   expert could guess on the MC propagation over the area with missing or corrupted satellite data.
417   Figure 7 demonstrates the characteristics of the best model (third-order ensemble-averaging
418   model) regarding false negatives (FN). Since the testing set is unbalanced with respect to stages, types
419   of cyclogenesis and cloud vortex types, we present in Figure 7acd relative FN rates for each separate
420   class in each taxonomy. We present the testing set distribution of classes for these taxonomies as well.
421   Note that scales are different for reference distributions of classes of the testing set and the
422   distributions of missed MCs. Detailed false negatives characteristics may be found in Appendix D.
423

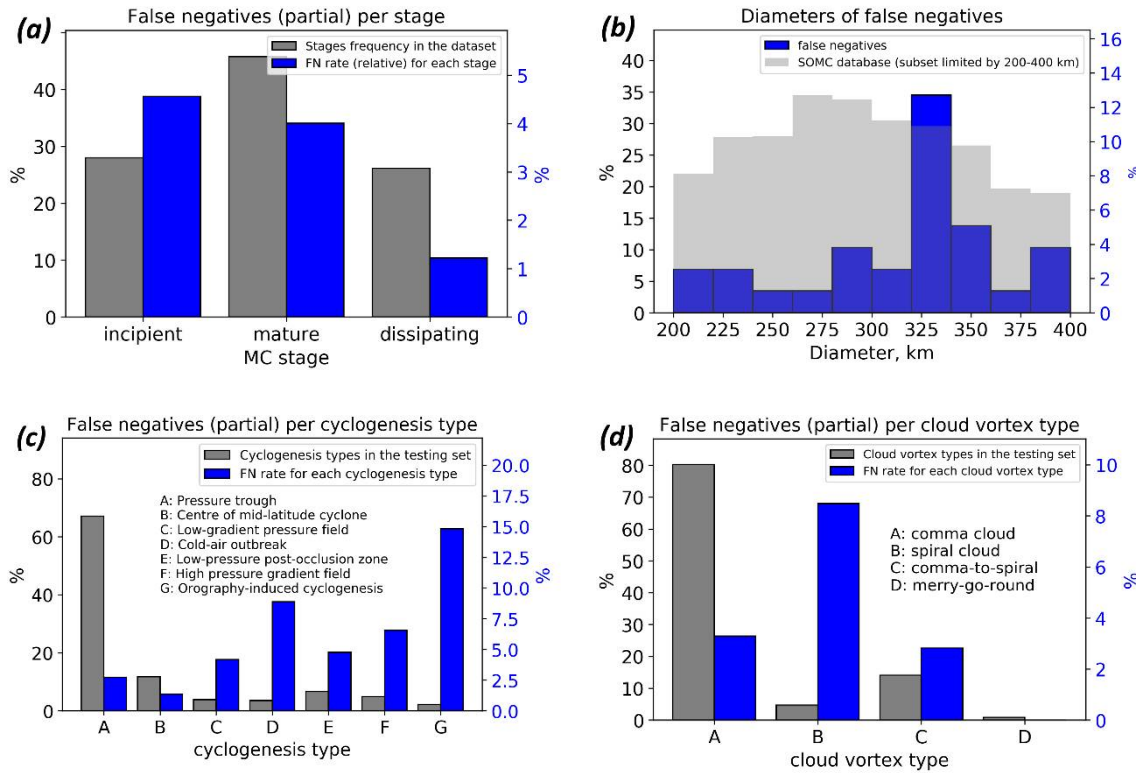



Figure 7. False negatives (FN, which are missed MCs) in the never-seen by the model testing set with
respect to (a) lifecycle stages; (b) diameters; (c) cyclogenesis types; (d) types of cloud vortex.

426   Tracking procedure requires the sustainable ability of the MCs detection scheme to recognize
427   mesocyclone cloud shape imprints during the whole MC life cycle. Figure 7a demonstrates that the
428   best model classifies mesocyclone imprints almost equally for incipient (~4.6% incipient missed) and
429   mature (~4% mature missed) stages. The fraction of missed MCs in its dissipating stage is lower (~4%
430   missed among MCs in dissipating stage). As for distribution of missed MCs with respect to their
431   diameters (see Fig. 7b), the histogram demonstrates fractions of FN objects relative to the whole FN
432   number. The distribution of MC diameters in the testing set in Figure 7b is shown as a reference.
433   There is a peak around the diameter value of 325 km, which does not coincide with any issues of
434   distributions of MC diameters when the testing set is subset by any particular class of any taxonomy.
435   However, since the total number of missed MCs is too small, there is no obvious reason to make
436   assumptions on the origin of this issue. The FN rates per cyclogenesis types (Fig. 7c) demonstrate the
437   only issue for the orography-induced MCs. This issue is caused by the total number of that
438   cyclogenesis type, which is small (only 27 MCs in the testing set and only 134 in the training set), so
439   the 4 which were missed is a substantial fraction of it. The same issue is demonstrated for the FN

440 rates per cloud vortex types. Since the total number of "spiral cloud" type in the testing set is
441 relatively small (59 of 1253), the 5 missed are a substantial fraction of it, compared to 33 missed of
442 1006 for "comma cloud" type.

443 **5. Conclusions and outlook**

444   In this study, we present an adaptation of a DCNN method resulting in an algorithm that
445 recognizes MCs signatures in preselected patches of satellite imagery of cloudiness and spatially
446 collocated WV imagery. The DCNN technique shows very high accuracy in this problem. The best
447 accuracy score of 97% is reached using the third-order ensemble-averaging model (6 models
448 ensemble) and the combination of both IR and WV images as input. We assess the accuracy of MCs
449 recognition by comparison of identified MCs (true/false - image contain MC/no MC on the image
450 parameter) with a reference dataset [6]. We demonstrate that deep convolutional networks are
451 capable of effectively detecting the presence of polar mesocyclone signatures in satellite imagery
452 patches of size 500 x 500 km. We also conclude that the quality of the satellite mosaics is sufficient
453 enough for performing the task of binary classification regarding the MCs presence in 500 x 500 km
454 patches, and for performing other similar tasks of pattern recognition type, e.g., semantic
455 segmentation of MCs.
456   Since the satellite-based studies of polar mesocyclone activity conducted in the SH (and in NH
457 as well) have never reported season-dependent variations of IR imprint of cloud shapes of MCs
458 [24,28,68,69], we assume the proposed methodology to be applicable to satellite imageries of polar
459 MCs available for the whole satellite observation era in SH. In the NH, the direct application of the
460 models that were trained on SH dataset is restricted due to the opposite sign of relative vorticity and
461 thus, different cloud shape orientation. However the proposed approach is still applicable, and the
462 only need is a dataset of tracks of MCs from the NH.
463   It was also shown that the accuracy of MCs detection by DCNNs is sensitive to the single (IR
464 only) or double (IR+WV) input data usage. IR+WV combination provides significant improvement of
465 the detection of MCs and allows a weak DCNN (CNN #2) to detect MCs with higher accuracy
466 compared to the weak CNN #1 (89.3% and 96.3% correspondingly). The computational cost of DCNN
467 training and hyper-parameters optimization for deep neural networks are time- and computational-
468 consuming. However, once trained, the computational cost of the DCNN inference is low.
469 Furthermore, the trained DCNN performs much faster compared to a human expert. Another
470 advantage of the proposed method is the low computational cost of data preprocessing that allows
471 the processing of satellite imagery in real time or the processing of large amounts of collected satellite
472 data.
473   We plan to extend the usage of this set of DCNNs (Table 2) for the development of an MCs
474 tracking method based on machine learning and using satellite IR and WV mosaics. These efforts
475 would be mainly focused on the development of the optimal choice of the "cut-off" window that has
476 to be applied to the satellite mosaic. In the case of a sliding-window approach (e.g., running the
477 500x500km sliding window through the mosaics), the virtual testing dataset of the whole mosaic is
478 highly unbalanced, so a model with non-zero FPR evaluated on balanced dataset would produce
479 much higher FPR. Thus, we expect the sliding-window approach not to be accurate enough in the
480 problem of MC detection. In the future, instead of the sliding-window, the Unet-like [70,71]
481 architecture should be considered with the binary semantic segmentation problem formulation. Since
482 the models that have been applied in this study (specifically their convolutional cores) are capable of
483 extracting the hidden representation which is relevant to MCs signatures, they may be used as the
484 encoder part of the Unet-like encoder-decoder neural network for MCs identification and tracking.
485 Considering MC tracking development, an approach proposed in a number of face recognition
486 studies should be reassuring [72,73]. This approach can be applied in a manner of triple-based
487 training of the DCNN to estimate a measure of similarity between one particular MC signatures in
488 consecutive satellite mosaics.

489

Author Contributions: Conceptualization, Mikhail Krinitskiy, Polina Verezemskaya and Sergey Gulev; Data curation, Mikhail Krinitskiy and Matthew Lazzara; Formal analysis, Mikhail Krinitskiy; Funding acquisition, Sergey Gulev; Investigation, Mikhail Krinitskiy and Kirill Grashchenkov; Methodology, Mikhail Krinitskiy and Polina Verezemskaya; Project administration, Mikhail Krinitskiy; Resources, Polina Verezemskaya and Sergey Gulev; Software, Mikhail Krinitskiy and Kirill Grashchenkov; Supervision, Sergey Gulev; Validation, Mikhail Krinitskiy, Polina Verezemskaya and Sergey Gulev; Visualization, Mikhail Krinitskiy and Polina Verezemskaya; Writing – original draft, Mikhail Krinitskiy, Polina Verezemskaya, Natalia Tilinina and Matthew Lazzara; Writing – review & editing, Natalia Tilinina, Sergey Gulev and Matthew Lazzara.

Funding: This research was funded by the Russian Ministry of Education and Science (agreement 14.613.21.0083, project ID RFMEFI61317X0083). Materials from MAL are based upon the work funded by the United States National Science Foundation under grants ANT-1244924 and ANT-1535632.

Acknowledgments: Computational resources for this research were provided by the Shared Facility Center "Data Center of FEB RAS", Khabarovsk, Russia.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Appendix A. DCNN best practices and additional techniques

There is a set of best practices commonly used to construct DCNNs for solving classification problems [56]. Modern DCNNs are built on the basis of consecutive convolutional and subsampling layers by performing nonlinear transformation of the initial data (see Fig. 2 in [42]). The primary layer type of convolutional neural networks (CNNs) is the so-called convolutional layer which is designed to extract visual patterns density map using discrete convolution operation with K (tends to be from 3 to 1000) kernels followed by a nonlinear transformation operation (activation function). One additional layer type is a pooling layer performing subsampling operation with one of the following aggregation functions: maximum, minimum, mean or others. In the current practice the maximum is used.

Since the LeNet DCNN [42] several studies [42–45] have demonstrated that the usage of consecutive convolutional and subsampling layers results in a skillful detection of various spatial patterns from the input 2D sample. The approach proposed in [42] implies the use of the output of these stacked layers set as an input data for a classifier, which in general may be any method suitable for classification problems, such as linear models, logistic regression, etc. LeCun [42] suggested to use the neural classifier, and this is now a conventional approach. The advantage of using a neural classifier is the ability to train the whole model at once (the so-called end-to-end training).

The whole model built in this manner represents a classifier capable of direct predicting a target value for the sample. We term the fully-connected (FC) layers set as "FC classifier", and the preceding part containing convolutional and pooling layers as "convolutional core" (see Figures 3,4).

For building a DCNN it is important to account for data dimensionality during its transformations from layer to layer. The input for a DCNN is an image represented by a matrix of the size (h, w, d), where h and w correspond to the image height and width in pixels, d is its levels number, the so-called depth (e.g., $d = 3$ when levels are red, green and blue channels of a colorful image). For the water vapor or radio-brightness temperature satellite data, $d = 1$. A convolutional layer and subsampling layer are described in details in [42]. Convolutional layers are characterized by their kernel sizes (e.g. 3x3, 5x5), their kernel numbers K and the nonlinear operation used (e.g. $tanh$ in [42]). Subsampling layers are characterized by their receptive field sizes e.g. 3x3, 5x5 etc. The output of a convolutional layer with K kernels is the so-called feature maps which is a matrix of the size (h, w, K). The nonlinear operation transforms it to a matrix of size $(h, w, 1)$. The following subsampling layer reduces the matrix size depending on the subsampling layer kernel size. Typically, this size is (2, 2) or (3, 3). Thus, the subsampling operation reduces the sample size by a factor 2 or 3, respectively. The output of a convolutional core is a set of abstract feature maps which is represented

540   by a 3D matrix. This matrix, being reshaped into a vector, is passed as the input to the FC classifier
541   (see Figures 3,4).

542   FC classifier of all models of this study includes hidden FC layers whose count varied
543   from 2 to 4. FC layers are characterized by the number of its basic building blocks (so-called artificial
544   neurons), which transform input data according to their rules (activation function) and parameters
545   (so-called weights) [53,74]. Artificial neurons count of FC1 which is the layer following the
546   convolutional core (see Figures 3,4), is chosen from the set {128, 256, 512, 1024}. The size of each
547   following FC layer is half of the preceding one, but not less than 128. The output layer is fully-
548   connected as well and contains one output node. For example, the structure of FC classifier in terms
549   of nodes count of layers might be the following: {512; 256; 128; 1}. All FC layers are alternated with
550   dropout layers in order to prevent overfitting of the model. All trainable layers' activation functions
551   are Rectified Linear Unit (ReLU):

$$\sigma_{ReLU}(z) = \max(0; z) \, , \tag{A1}$$

552   except the output layer whose activation function is sigmoid:

$$\sigma_{sigm}(z) = \frac{1}{1 + e^{-\theta z}} \, , \tag{A2}$$

553   where θ are layers' trainable parameters.

554   In order to measure the error of the network on each individual sample during the training
555   process we use the binary cross-entropy as a loss function:

$$\mathcal{L} = \sum_{i=0}^{N} (y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)) \, , \tag{A3}$$

556   where y_i is the expert-defined ground truth for the target value, $\hat{y}_i$ is the estimated probability of
557   the i-th sample to be true, N is samples count of the training set or a training mini-batch. This loss
558   function is minimized in the space of the model weights using the method of backpropagation of
559   error [66] denoted as "backprop training" in Figures 3,4. The outcome of the the whole model is the
560   probability of each class for the input sample. In the case of binary classification, the FC classifier has
561   one output unit, producing probability of MC presence for the input sample.

562

563   In addition to the basic approach proposed in [42] a number of techniques may be applied. Using
564   them one can construct and train DCNNs of various accuracy and various generalization abilities
565   which is characterized by the quality of a model estimated on a never-seen test data.

566   *A.1. Transfer learning*

567   One of the additional approaches is Transfer Learning (TL) [57–62]. Generally, this technique
568   focuses on storing the knowledge obtained by some network while being trained for one problem
569   and applying it to another problem of a similar kind. In practice, this approach implies the DCNN
570   structure to be built using some part of a network previously trained on a considerable amount of
571   data, for example, ImageNet [49]. In these terms, VGG16 [1] is not only an efficient architecture, but
572   also the pre-trained network containing optimized weights values (also known as network
573   parameters). Best practice for building a new advanced DCNN based on TL approach is to compose
574   it using convolutional core of the pre-trained model (e.g. VGG16) followed by a new FC neural
575   classifier. Weights of the convolutional part in this case are fixed, and only FC part is optimized. In
576   this approach, the convolutional core may be considered as a feature extractor (see [42]), which
577   computes a highly relevant low-dimensional (compared to original samples dimensionality) vector,
578   representing the data (e.g. "reshaped to vector" output of the convolutional core in Fig. 3).

579   *A.2. Fine Tuning*

580   TL approach relies on the similarity of data distributions within two datasets. But in the case of
581   significant differences, for example in terms of Kullback–Leibler divergence between some particular

582   feature approximated probability distributions, the new FC classifier capabilities may not cover all
583   of those differences. In this case, some layers of the convolutional core, that are close to FC classifier,
584   can be turned on to be optimized (the so-called Fine Tuning, FT). Regarding DCNNs application to
585   satellite mosaics, we have to consider that VGG16 was optimized on ImageNet dataset which
586   contains everyday-observed objects like buildings, dogs, cats, cars etc., without any satellite
587   imageries or even clouds. So FT approach can be considered as a promising approach when
588   composing MC-detecting DCNN at IR and WV satellite mosaic data.

589   *A.3. Preventing overfitting*

590   Machine learning models and neural networks, in particular, may vary in terms of complexity.
591   Overly complex models may suffer from overfitting: the effect of poor target prediction quality on
592   unseen data concurrently with nearly exact prediction of target values on training data. There are
593   several state-of-the-art approaches to prevent overfitting of neural networks. We used most fruitful
594   and reliable ones: dropout [64] and data augmentation also called auxiliary variables [65]. We also
595   used ensemble averaging of the models outcome.

596   *A.4. Preventing overfitting with dropout*

597   Dropout approach is the way of preventing overfit with a computationally inexpensive but still
598   powerful method of regularizing neural networks through bagging [75] and virtually ensembling
599   models of similar architecture. Bagging involves training multiple models and testing each of them
600   on test samples. Since training and evaluating of deep neural networks tend to be time-consuming
601   and computationally expensive, the original bagging approach [75] seems to be impractical. With the
602   dropout approach applied, the network may be thought of as an ensemble of all sub-networks that
603   can be composed by removing non-output nodes from the base network. In practice, this approach
604   is implemented by dropout layer which turns the preceding layer output to zero for each node with
605   some probability p. This procedure repeats for each mini-batch at the training time. At the inference
606   time, the dropout approach involves network weights scaling by $1/p$. Each of our models includes
607   dropout layers between trainable layers. Rate p was set to 0.1 for each dropout layer of each model.

608   *A.5. Preventing overfitting with dataset augmentation*

609   Dataset augmentation is the state-of-the-art way to make a machine learning model generalize
610   better. When available dataset size is limited, the way to get around that is to generate fake data
611   which should be similar to real samples. Best practice for DCNNs is generating fake samples by
612   adding some noise or applying slight transformations like shift, shear, rotation, scaling etc. Formally,
613   with data augmentation one can increase variability of features of the original dataset and
614   substantially extend its size. This approach often improves generalization ability of the trained
615   model.
616   We trained each of our models with data augmentation approach applied. The rotation angle
617   range was 90° in both direction; independent width and height scaling performed within range from
618   0.8 to 1.2; zoom range from 0.8 to 1.2; shear angle range from -2° to 2°. We did not use flipping
619   upside-down and left-to-right.

620   *A.6. Preventing overfitting with ensemble averaging*

621   In general, during the parameters optimization (learning process) each DCNN converges to a
622   local minimum of the loss function in the space of its weights. The training process starts from a
623   randomly generated point of this space. Due to a non-convexity of loss function, every new DCNN
624   model converges to a new local minimum. Some models may converge to a minimum that is not
625   really close to a global one in terms of loss function value, and thus the quality measure of that model
626   remains poor. Other models may converge to a good minimum that is close to a global one in terms
627   of loss function value, but this proximity may lead to a poor generalization ability which means low
628   quality measure estimated on a testing subset of data. There are approaches for improving the

629 generalization ability of several models that are generally similar, but differ in detailed predictions.
630 In our study we applied simple ensemble averaging [76], which is one of state-of-the-art approaches
631 for improving machine learning models generalization ability. With this approach several models of
632 each architecture are trained, and probabilities of these models are averaged. The prediction of this
633 model is treated as an ensemble outcome:

$$p_i = \frac{\sum_{m=0}^{M} p_i^{(m)}}{M} , \tag{A4}$$

634 where p_i is the estimated probability of the ensemble of M models for i-th sample to be true; each
635 m-th model`s probability estimation for i-th sample to be true is $p_i^{(m)}$. In this study we applied
636 ensembling on DCNNs of identical architectures. The resulting models we term _second-order models_
637 in this study. They are synthetic ones that are not trained, but are ensembles.
638   Satellite IR+WV snapshots or satellite IR snapshot alone are essentially the object description,
639 and each model that is presented in our study produces the outcome for each object regardless of the
640 description - whether it is IR snapshot alone or IR+WV snapshots. So there is an opportunity to
641 average probability outcomes of all the models of this study. The resulting model that produces
642 averaged probabilities of the ensemble containing all trained models we term _third-order model_. It is a
643 synthetic one that is not trained, but is an ensemble.

644 _A.7. Adjustment of the probability threshold_

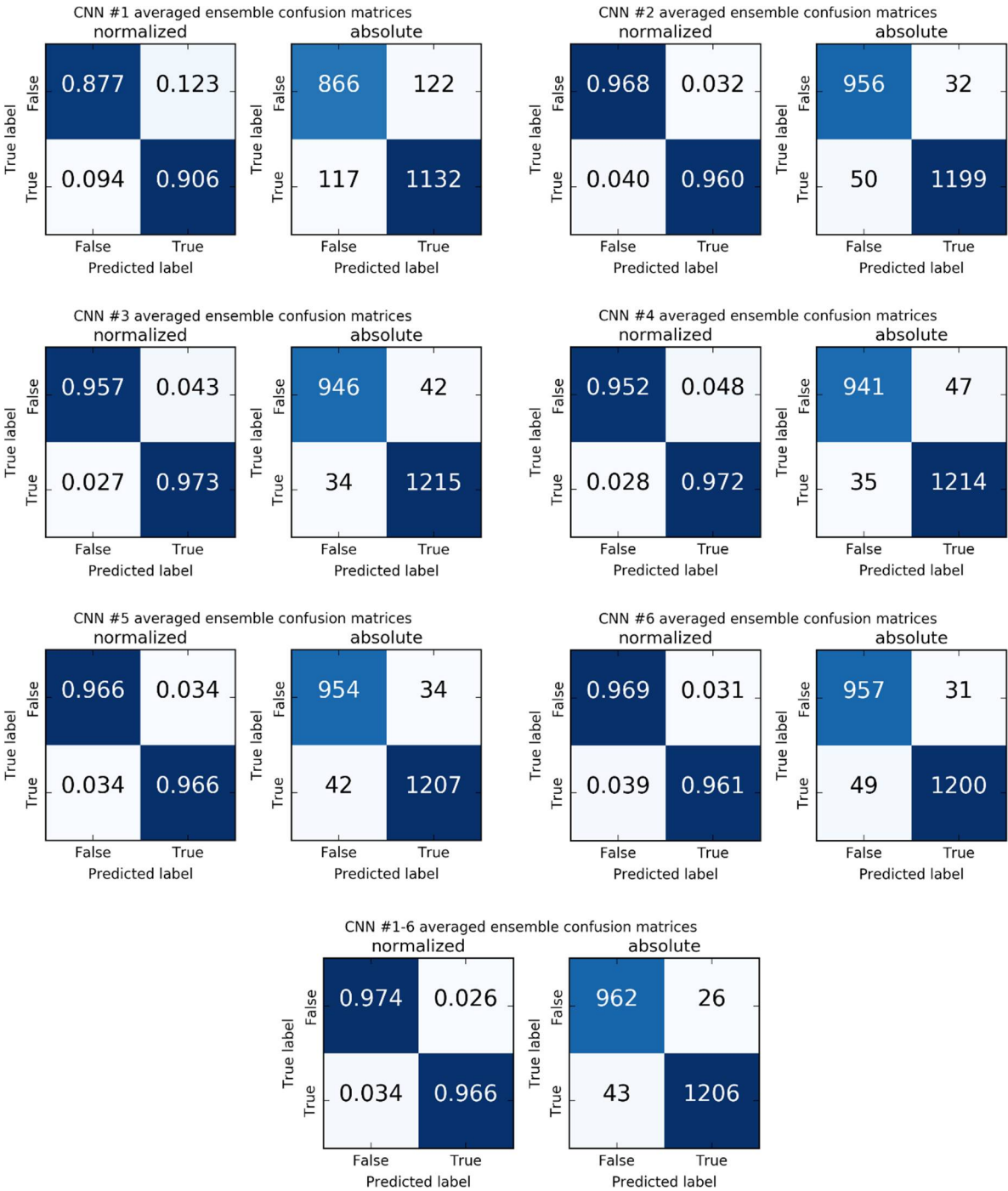
645   The outcome of each model of this study is the estimation of the probability for the sample to be
646 true (i.e. to contain an MC). So there is an arbitrariness in choosing the threshold of this probability
647 to get the outcome which is binary. The most common way to choose this threshold is the ROC curve
648 analysis. Each point of this curve represents the False Positive Rate (FPR) and True Positive Rate
649 (TPR) combination for the particular probability threshold p_{th} (e.g. see Fig. 6bdf). The model
650 performing true random choice between true and false outcome has a ROC curve on the main
651 diagonal of this plot. The ROC curve of the perfect classifier follows from the point (0.0, 0.0) straight
652 to the point (0.0, 1.0) and then to the point (1.0, 1.0). The area under the ROC curve (AUC ROC) may
653 be considered as a measure of model quality. The best model AUC ROC is 1.0, the true random choice
654 model AUC ROC is 0.5, and the worst model AUC ROC is 0.0.
655   In a range of cases the best accuracy score might not be reached with $p_{th} = 0.5$. The lines of equal
656 accuracy score, as presented in Fig. 6bdf, are diagonal. In case of perfect 50/50 ratio of true/false
657 samples they are parallel to the main diagonal. In case of slight inequality of true and false samples
658 count these lines have slightly different slope as shown in Fig. 6bdf. For each accuracy score there are
659 two, one or no points of the ROC curve intersection with the accuracy isoline. So if a model is
660 represented with a ROC curve, the maximum value of its Acc is located at the point of this curve
661 where the accuracy isoline is tangent to it. For each model of this study including second- and third-
662 order models, the optimal probability threshold was estimated based on ROC curve analysis.
663

664   **Appendix B. CNN #3 and CNN #5 Best hyper-parameters combinations.**

665   According to section 3.4, CNN #3 and CNN #5 are both constructed to have one-branched
666   convolutional core. Best combination of hyper-parameters of these networks are the same. The only
667   difference is the FT approach that was applied in case of CNN #5.
668
669   **Table B1.** CNN #3 and CNN #5 best hyper-parameters combination.

Layer (block) name	Layer (block) nodes count or output dimensions	Connected to
Input_data_IR	100x100	-
VGG_16_conv_core	see [1]; output: 3x3x512	Input_data_IR
Reshape_1	4608	VGG_16_conv_core
Dropout_1	4608	Reshape_1
FC1	1024	Dropout_1
Dropout_2	1024	FC1
FC2	512	Dropout_2
Dropout_3	512	FC2
FC3	256	Dropout_3
Dropout_4	256	FC3
FC4	128	Dropout_4
FC_output	1	FC3

670
671

672   **Appendix C. Detailed performance metrics of all DCNN models.**

673   **Figure C1.** Confusion matrices for all models and the third-order model CNN #1-6 averaged
674   ensemble, computed on test never-seen subset of data. For each architecture the ensemble averaging
675   technique is applied.

676

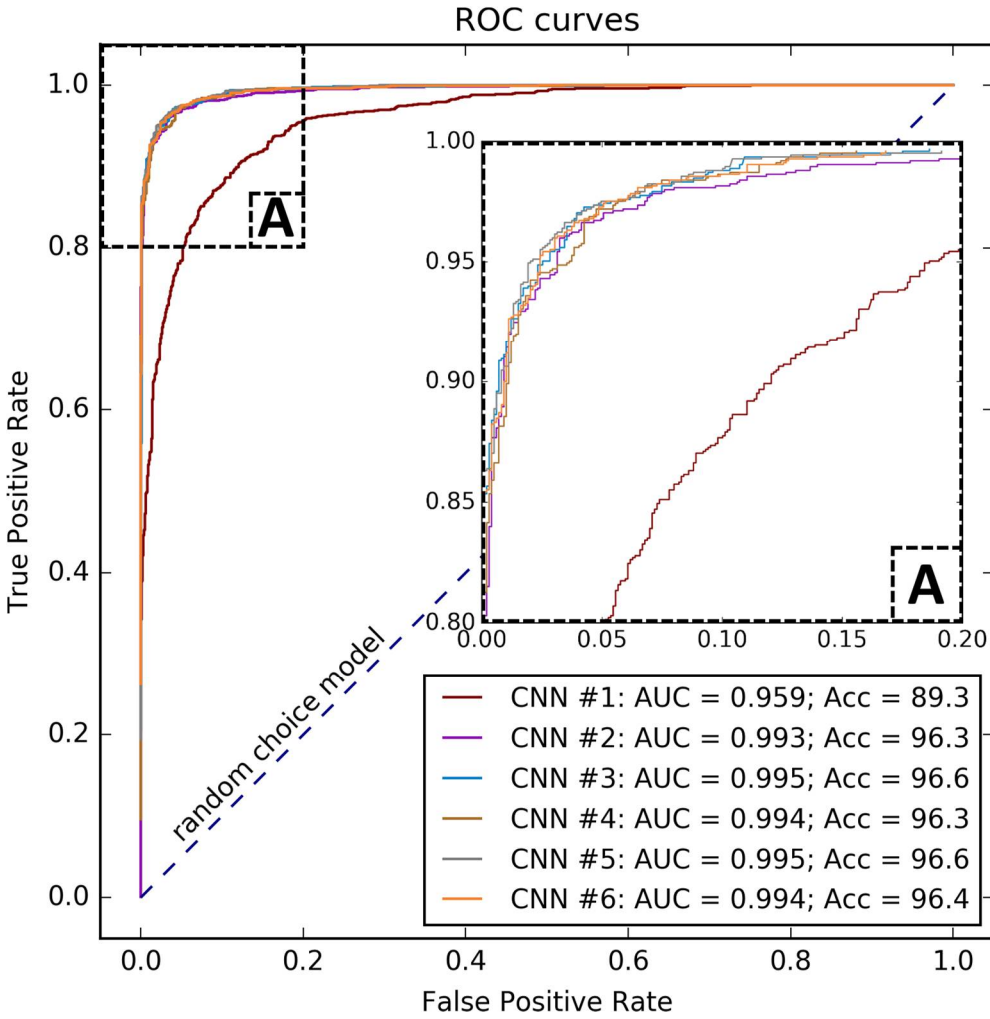



Figure C2. Receiver operating characteristic curves computed on test never-seen subset of data for all models. For each architecture the ensemble averaging technique is applied.

680   **Appendix D. Detailed false negative rates of the third-order ensemble-averaging model.**

681   **Table D1.** False negative rates per cyclogenesis types.

Cyclogenesis type	Testing set, objects number	False negatives, objects number	FN relative rate, %
Pressure trough	841	23	2.7
Centre of mid-latitude cyclone	147	2	1.4
Low-gradient pressure field	48	2	4.2
Cold-air outbreak	45	4	8.9
Low-pressure post-occlusion zone	84	4	4.8
High pressure gradient field	61	4	6.6
Orography-induced cyclogenesis	27	4	14.8

682

683   **Table D2.** False negative rates per cloud vortex types.

Cloud vortex type	Testing set, objects number	False negatives, objects number	FN relative rate, %
Comma cloud	1006	33	3.3
Spiral cloud	59	5	8.5
Comma-to-spiral	177	5	2.3
Merry-go-round	11	0	0.0

684

685   **Table D3.** False negative rates per MC stages.

MC stage	Testing set, objects number	False negatives, objects number	FN relative rate, %
Incipient	352	16	4.6
Mature	574	23	4.0
Dissipating	327	4	1.2

686

687   **References**

688   1.   Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition.
689   *arXiv:1409.1556 [cs]* **2014**.

690   2.   *Polar Lows: Mesoscale Weather Systems in the Polar Regions*; Rasmussen, E.A., Turner, J., Eds.; Cambridge
691   University Press: Cambridge, 2003; ISBN 978-0-511-52497-4.

692   3.   Marshall, J.; Schott, F. Open-ocean convection: Observations, theory, and models. *Reviews of Geophysics*
693   **1999**, *37*, 1–64, doi:10.1029/98RG02739.

694   4.   Condron, A.; Renfrew, I.A. The impact of polar mesoscale storms on northeast Atlantic Ocean circulation.
695   *Nature Geoscience* **2013**, *6*, 34–37, doi:10.1038/ngeo1661.

696   5.   Condron, A.; Bigg, G.R.; Renfrew, I.A. Modeling the impact of polar mesocyclones on ocean circulation.
697   *Journal of Geophysical Research: Oceans* **2008**, *113*, doi:10.1029/2007JC004599.

698   6.   Verezemskaya, P.; Tilinina, N.; Gulev, S.; Renfrew, I.A.; Lazzara, M. Southern Ocean mesocyclones and
699   polar lows from manually tracked satellite mosaics. *Geophysical Research Letters* **2017**, *44*, 7985–7993,
700   doi:10.1002/2017GL074053.

701 7.   Laffineur, T.; Claud, C.; Chaboureau, J.-P.; Noer, G. Polar Lows over the Nordic Seas: Improved
702   Representation in ERA-Interim Compared to ERA-40 and the Impact on Downscaled Simulations. *Mon.*
703   *Wea. Rev.* **2014**, *142*, 2271–2289, doi:10.1175/MWR-D-13-00171.1.

704 8.   Michel, C.; Terpstra, A.; Spengler, T. Polar Mesoscale Cyclone Climatology for the Nordic Seas Based on
705   ERA-Interim. *J. Climate* **2017**, *31*, 2511–2532, doi:10.1175/JCLI-D-16-0890.1.

706 9.   Bromwich, D.H.; Wilson, A.B.; Bai, L.-S.; Moore, G.W.K.; Bauer, P. A comparison of the regional Arctic
707   System Reanalysis and the global ERA-Interim Reanalysis for the Arctic. *Quarterly Journal of the Royal*
708   *Meteorological Society* **2016**, *142*, 644–658, doi:10.1002/qj.2527.

709 10.   Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. *Numerical recipes 3rd edition: The art of scientific*
710   *computing*; Cambridge university press, 2007; ISBN 0-521-88068-8.

711 11.   Rojo, M.; Claud, C.; Mallet, P.-E.; Noer, G.; Carleton, A.M.; Vicomte, M. Polar low tracks over the Nordic
712   Seas: a 14-winter climatic analysis. *Tellus A: Dynamic Meteorology and Oceanography* **2015**, *67*, 24660,
713   doi:10.3402/tellusa.v67.24660.

714 12.   Smirnova, J.; Golubkin, P. Comparing Polar Lows in Atmospheric Reanalyses: Arctic System Reanalysis
715   versus ERA-Interim. *Mon. Wea. Rev.* **2017**, *145*, 2375–2383, doi:10.1175/MWR-D-16-0333.1.

716 13.   Hines, K.M.; Bromwich, D.H. Development and Testing of Polar Weather Research and Forecasting (WRF)
717   Model. Part I: Greenland Ice Sheet Meteorology. *Mon. Wea. Rev.* **2008**, *136*, 1971–1989,
718   doi:10.1175/2007MWR2112.1.

719 14.   Zappa, G.; Shaffrey, L.; Hodges, K. Can Polar Lows be Objectively Identified and Tracked in the ECMWF
720   Operational Analysis and the ERA-Interim Reanalysis? *Mon. Wea. Rev.* **2014**, *142*, 2596–2608,
721   doi:10.1175/MWR-D-14-00064.1.

722 15.   Pezza, A.; Sadler, K.; Uotila, P.; Vihma, T.; Mesquita, M.D.S.; Reid, P. Southern Hemisphere strong polar
723   mesoscale cyclones in high-resolution datasets. *Clim Dyn* **2016**, *47*, 1647–1660, doi:10.1007/s00382-015-
724   2925-2.

725 16.   Xia, L.; Zahn, M.; Hodges, K.; Feser, F.; Storch, H. A comparison of two identification and tracking
726   methods for polar lows. *Tellus A: Dynamic Meteorology and Oceanography* **2012**, *64*, 17196,
727   doi:10.3402/tellusa.v64i0.17196.

728 17.   Harold, J.M.; Bigg, G.R.; Turner, J. Mesocyclone activity over the North-East Atlantic. Part 1: vortex
729   distribution and variability. *International Journal of Climatology* **1999**, *19*, 1187–1204, doi:10.1002/(SICI)1097-
730   0088(199909)19:11<1187::AID-JOC419>3.0.CO;2-Q.

731 18.   Noer, G.; Saetra, Ø.; Lien, T.; Gusdal, Y. A climatological study of polar lows in the Nordic Seas. *Quarterly*
732   *Journal of the Royal Meteorological Society* **2011**, *137*, 1762–1772, doi:10.1002/qj.846.

733 19.   Smirnova, J.E.; Zabolotskikh, E.V.; Bobylev, L.P.; Chapron, B. Statistical characteristics of polar lows over
734   the Nordic Seas based on satellite passive microwave data. *Izv. Atmos. Ocean. Phys.* **2016**, *52*, 1128–1136,
735   doi:10.1134/S0001433816090255.

736 20.   McMurdie, L.A.; Claud, C.; Atakturk, S. Satellite-derived atmospheric characteristics of spiral and comma-
737   shaped southern hemisphere mesocyclones. *Journal of Geophysical Research: Atmospheres* **1997**, *102*, 13889–
738   13905, doi:10.1029/97JD00279.

739 21.   Irving, D.; Simmonds, I.; Keay, K. Mesoscale Cyclone Activity over the Ice-Free Southern Ocean: 1999–
740   2008. *J. Climate* **2010**, *23*, 5404–5420, doi:10.1175/2010JCLI3628.1.

741 22.   Neu, U.; Akperov, M.G.; Bellenbaum, N.; Benestad, R.; Blender, R.; Caballero, R.; Cocozza, A.; Dacre, H.F.;
742   Feng, Y.; Fraedrich, K.; Grieger, J.; Gulev, S.; Hanley, J.; Hewson, T.; Inatsu, M.; Keay, K.; Kew, S.F.;
743   Kindem, I.; Leckebusch, G.C.; Liberato, M.L.R.; Lionello, P.; Mokhov, I.I.; Pinto, J.G.; Raible, C.C.; Reale,

744   M.; Rudeva, I.; Schuster, M.; Simmonds, I.; Sinclair, M.; Sprenger, M.; Tilinina, N.D.; Trigo, I.F.; Ulbrich,
745   S.; Ulbrich, U.; Wang, X.L.; Wernli, H. IMILAST: A Community Effort to Intercompare Extratropical
746   Cyclone Detection and Tracking Algorithms. *Bull. Amer. Meteor. Soc.* **2013**, *94*, 529–547, doi:10.1175/BAMS-
747   D-11-00154.1.

748   23. Wilhelmsen, K. Climatological study of gale-producing polar lows near Norway. *Tellus A: Dynamic*
749   *Meteorology and Oceanography* **1985**, *37*, 451–459, doi:10.3402/tellusa.v37i5.11688.

750   24. Carrasco, J.F.; Bromwich, D.H. Mesoscale cyclogenesis dynamics over the southwestern Ross Sea,
751   Antarctica. *Journal of Geophysical Research: Atmospheres* **1993**, *98*, 12973–12995.

752   25. Carrasco, J.F.; Bromwich, D.H.; Liu, Z. Mesoscale cyclone activity over Antarctica during 1991: 1. Marie
753   Byrd Land. *Journal of Geophysical Research: Atmospheres* **1997**, *102*, 13923–13937, doi:10.1029/97JD00905.

754   26. Turner, J.; Thomas, J.P. Summer-season mesoscale cyclones in the bellingshausen-weddell region of the
755   antarctic and links with the synoptic-scale environment. *International Journal of Climatology* **1994**, *14*, 871–
756   894, doi:10.1002/joc.3370140805.

757   27. Harold, J.M.; Bigg, G.R.; Turner, J. Mesocyclone activity over the Northeast Atlantic. Part 2: An
758   investigation of causal mechanisms. *International Journal of Climatology* **1999**, *19*, 1283–1299,
759   doi:10.1002/(SICI)1097-0088(199910)19:12<1283::AID-JOC420>3.0.CO;2-T.

760   28. CARLETON, A.M. On the interpretation and classification of mesoscale cyclones from satellite infrared
761   imagery. *International Journal of Remote Sensing* **1995**, *16*, 2457–2485, doi:10.1080/01431169508954569.

762   29. Claud, C.; Katsaros, K.B.; Mognard, N.M.; Scott, N.A. Comparative satellite study of mesoscale
763   disturbances in polar regions. *Global Atmos Ocean Syst* **1996**, *4*, 233–273.

764   30. Gang, F.; Qin-yu, L.; Zeng-mao, W. General features of polar lows over the Japan Sea and the
765   Northwestern Pacific. *Chin. J. Ocean. Limnol.* **1999**, *17*, 300–307, doi:10.1007/BF02842823.

766   31. Gurvich, I.A.; Pichugin, M.K. Study of the compara tive characteristics of typical mesoscale cyclones over
767   Far Eastern seas on the basis of satellite multisensory sounding. *Sovrem. Probl. Distantsionnogo*
768   *Zondirovaniya Zemli Kosmosa* **2013**, *10*, 51–59.

769   32. Claud, C.; Carleton, A.M.; Duchiron, B.; Terray, P. Southern hemisphere winter cold-air mesocyclones:
770   climatic environments and associations with teleconnections. *Climate Dynamics* **2009**, *33*, 383–408,
771   doi:10.1007/s00382-008-0468-5.

772   33. Blechschmidt, A.-M. A 2-year climatology of polar low events over the Nordic Seas from satellite remote
773   sensing. *Geophysical Research Letters* **2008**, *35*, doi:10.1029/2008GL033706.

774   34. Heinle, A.; Macke, A.; Srivastav, A. Automatic cloud classification of whole sky images. *Atmospheric*
775   *Measurement Techniques* **2010**, *3*, 557–567, doi:10.5194/amt-3-557-2010.

776   35. Taravat, A.; Frate, F.D.; Cornaro, C.; Vergari, S. Neural Networks and Support Vector Machine Algorithms
777   for Automatic Cloud Classification of Whole-Sky Ground-Based Images. *IEEE Geoscience and Remote*
778   *Sensing Letters* **2015**, *12*, 666–670, doi:10.1109/LGRS.2014.2356616.

779   36. Onishi, R.; Sugiyama, D. Deep Convolutional Neural Network for Cloud Coverage Estimation from
780   Snapshot Camera Images. *SOLA* **2017**, *13*, 235–239, doi:10.2151/sola.2017-043.

781   37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural
782   networks. In *Advances in neural information processing systems*; 2012; pp. 1097–1105.

783   38. Shin, H.-C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep
784   Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset
785   Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging* **2016**, *35*, 1285–1298,
786   doi:10.1109/TMI.2016.2528162.

787 39. Krinitskiy, M.A. Application of machine learning methods to the solar disk state detection by all-sky
788   images over the ocean. *Oceanology* **2017**, *57*, 265–269, doi:10.1134/S0001437017020126.

789 40. Liu, Y.; Racah, E.; Correa, J.; Khosrowshahi, A.; Lavers, D.; Kunkel, K.; Wehner, M.; Collins, W.
790   Application of deep convolutional neural networks for detecting extreme weather in climate datasets.
791   *arXiv preprint arXiv:1605.01156* **2016**.

792 41. Huang, D.; Du, Y.; He, Q.; Song, W.; Liotta, A. DeepEddy: A simple deep architecture for mesoscale
793   oceanic eddy detection in SAR images. In *2017 IEEE 14th International Conference on Networking, Sensing*
794   *and Control (ICNSC)*; 2017; pp. 673–678.

795 42. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition.
796   *Proceedings of the IEEE* **1998**, *86*, 2278–2324, doi:10.1109/5.726791.

797 43. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444, doi:10.1038/nature14539.

798 44. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures
799   and their applications. *Neurocomputing* **2017**, *234*, 11–26, doi:10.1016/j.neucom.2016.12.038.

800 45. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A
801   review. *Neurocomputing* **2016**, *187*, 27–48, doi:10.1016/j.neucom.2015.09.116.

802 46. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich,
803   A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern*
804   *recognition*; 2015; pp. 1–9.

805 47. Chollet, F. Xception: Deep learning with depthwise separable convolutions, CoRR abs/1610.02357. *URL*
806   *http://arxiv. org/abs/1610.02357* **2016**.

807 48. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE*
808   *conference on computer vision and pattern recognition*; 2016; pp. 770–778.

809 49. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image
810   database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*; Ieee, 2009; pp.
811   248–255.

812 50. Eckersley, P.; Nasser, Y. AI Progress Measurement Available online: https://www.eff.org/ai/metrics
813   (accessed on Aug 13, 2018).

814 51. Deng, L.; Yu, D. Deep Learning: Methods and Applications. *SIG* **2014**, *7*, 197–387, doi:10.1561/2000000039.

815 52. Deng, L. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA*
816   *Transactions on Signal and Information Processing* **2014**, *3*, doi:10.1017/atsip.2013.9.

817 53. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural networks* **2015**, *61*, 85–117.

818 54. Lazzara, M.A.; Keller, L.M.; Stearns, C.R.; Thom, J.E.; Weidner, G.A. Antarctic satellite meteorology:
819   applications for weather forecasting. *Monthly Weather Review* **2003**, *131*, 371–383.

820 55. Kohrs, R.A.; Lazzara, M.A.; Robaidek, J.O.; Santek, D.A.; Knuth, S.L. Global satellite composites — 20
821   years of evolution. *Atmospheric Research* **2014**, *135–136*, 8–34, doi:10.1016/j.atmosres.2013.07.023.

822 56. Simard, P.Y.; Steinkraus, D.; Platt, J.C. Best practices for convolutional neural networks applied to visual
823   document analysis. In *Proceedings of Seventh International Conference on Document Analysis and Recognition*;
824   IEEE: Edinburgh, Scotland, 2003; p. 958.

825 57. Pratt, L.Y.; Mostow, J.; Kamm, C.A.; Kamm, A.A. Direct Transfer of Learned Information Among Neural
826   Networks. In *AAAI*; 1991; Vol. 91, pp. 584–589.

827 58. Caruana, R. Learning Many Related Tasks at the Same Time with Backpropagation. *Advances in neural*
828   *information processing systems* **1995**, 8.

829 59.   Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks
830   with multitask learning. In *Proceedings of the 25th international conference on Machine learning*; ACM, 2008;
831   pp. 160–167.

832 60.   Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*
833   **2010**, *22*, 1345–1359, doi:10.1109/TKDE.2009.191.

834 61.   Mesnil, G.; Dauphin, Y.; Glorot, X.; Rifai, S.; Bengio, Y.; Goodfellow, I.; Lavoie, E.; Muller, X.; Desjardins,
835   G.; Warde-Farley, D. Unsupervised and transfer learning challenge: a deep learning approach. In
836   *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop-Volume 27*;
837   JMLR. org, 2011; pp. 97–111.

838 62.   Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and Transferring Mid-Level Image Representations
839   using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern*
840   *Recognition*; 2014; pp. 1717–1724.

841 63.   Maclin, R.; Shavlik, J.W. Combining the predictions of multiple classifiers: Using competitive learning to
842   initialize neural networks. In *Proceedings of the 1995 International Joint Conference on AI*; Citeseer: Montreal,
843   Quebec, Canada, 1995; pp. 524–531.

844 64.   Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to
845   prevent neural networks from overfitting. *The Journal of Machine Learning Research* **2014**, *15*, 1929–1958.

846 65.   Agakov, F.V.; Barber, D. An Auxiliary Variational Method. In *Neural Information Processing*; Lecture Notes
847   in Computer Science; Springer, Berlin, Heidelberg, 2004; pp. 561–566.

848 66.   Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature*
849   **1986**, *323*, 533–536, doi:10.1038/323533a0.

850 67.   Sorokin, A.A.; Makogonov, S.V.; Korolev, S.P. The Information Infrastructure for Collective Scientific
851   Work in the Far East of Russia. *Sci. Tech. Inf. Proc.* **2017**, *44*, 302–304, doi:10.3103/S0147688217040153.

852 68.   Carleton, A.M.; Carpenter, D.A. Satellite climatology of 'polar lows' and broadscale climatic associations
853   for the Southern Hemisphere. *International Journal of Climatology* **1990**, *10*, 219–246,
854   doi:10.1002/joc.3370100302.

855 69.   Carleton, A.M.; Fitch, M. Synoptic aspects of Antarctic mesocyclones. *Journal of Geophysical Research:*
856   *Atmospheres* **1993**, *98*, 12997–13018, doi:10.1029/92JD02132.

857 70.   Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation.
858   In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*; Lecture Notes in Computer
859   Science; Springer, Cham, 2015; pp. 234–241.

860 71.   Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture
861   for Image Segmentation. *arXiv:1511.00561 [cs]* **2015**.

862 72.   Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep face recognition. In *BMVC*; 2015; Vol. 1, p. 6.

863 73.   Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and
864   Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2015; pp. 815–
865   823.

866 74.   Minsky, M.; Papert, S.A. *Perceptrons*; MIT Press: Cambridge, 1969; ISBN 978-0-262-13043-1.

867 75.   Breiman, L. Bagging predictors. *Mach Learn* **1996**, *24*, 123–140, doi:10.1007/BF00058655.

868 76.   Lincoln, W.P.; Skrzypek, J. Synergy of clustering multiple back propagation networks. In *Advances in*
869   *neural information processing systems*; 1990; pp. 650–657.

870

871