Article

Deep convolutional neural networks for detection of polar mesocyclones from satellite mosaics

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

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

Abstract: Polar mesocyclones (MCs) are small in size marine atmospheric phenomena accompanied by extremely strong surface winds and heat fluxes and thus largely influencing deep ocean water formation in the polar regions. Accurate detection of polar mesocyclones in high-resolution satellite data, while challenging, is a time-consuming task, when performed manually. Existing algorithms for the automatic detection of polar mesocyclones are based on the conventional analysis of patterns of cloudiness and involve different empirically defined thresholds of geophysical variables. As a result, different detection methods typically reveal very different results when applied to a single dataset. We present a conceptually novel approach for the detection of MCs based upon the use of deep convolutional neural networks (DCNNs). The training dataset is based on the reference database of manually tracked from satellite mosaics MCs in the Southern Hemisphere. This dataset is further used for testing several different setups of DCNN, specifically, DCNN "from scratch", DCNN based on VGG16 pre-trained weights engaging also the Transfer Learning technique, and DCNN based on VGG16 with Fine Tuning technique. Each of these networks is further applied to both IR and IR+WV satellite imagery. The best skills (97% of the binary classification accuracy score) is achieved with DCNN based on VGG16 pre-trained weights with both Transfer Learning and Fine Tuning techniques applied. The algorithm can be further extended to the automatic identification and tracking numerical scheme and applied to the other atmospheric phenomena characterized by a distinct signature on satellite imagery.

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

---

Nomenclature

BCE - binary cross-entropy
CNN - convolutional neural network
DCNN - deep convolutional neural network
DL - deep learning
FC - fully-connected
FNR - false negative rate
FPR - false positive rate
IR - infrared
MC - mesocyclone
NH - Northern Hemisphere
ROC - receiver operator characteristic
AUC ROC - area under the curve of receiver operator characteristic

 doi:10.20944/preprints201809.0361.v1

SH - Southern Hemisphere
SOMC - Shirshov Institute of Oceanology mesocyclone dataset for Southern Ocean
TNR - true negative rate
TPR - true positive rate
WV - integrated water vapor

1. Introduction

Polar mesoscale cyclones (MCs) are intense high-latitude marine atmospheric vortices. Their sizes range from 200 to 1000 km with the lifetimes spanning from 6 to 36 hours [1]. Specific type of mesocyclones (the so-called polar lows, PLs) is characterized by the surface wind of more than 15 m/s and strong surface fluxes. These PLs have a significant impact on the local weather conditions causing rough sea. Being relatively small in size (compared to the extratropical cyclones), MCs contribute significantly to the generation of extreme air-sea fluxes and initialize intense surface transformation of water masses resulting in the formation of ocean deep waters [2–4]. These processes are most intense in the Weddel and Bellingshausen Seas in the Southern hemisphere and in the Labrador, Greenland, Norway and Barents Seas in the Northern Hemisphere.

Being critically important for many oceanographic and meteorological applications, MCs are hardly detectable in different reanalysis datasets, mostly due to inadequate resolution of the products.

The spatial resolution of the modern reanalyses still does not MCs permit for the accurate identification of MCs. In [5] it is argued for at least 10 by 10 grid points necessary for effective capturing the MC. This implies about 30 km spatial resolution in the model or reanalysis for detecting MC with the diameter of 300 km. However, in [6] demonstrated that 48% of MCs (including PLs) in the SH are characterized by the diameters smaller than 300 km. Thus, even the latest very high-resolution ERA5 reanalysis [7,8] with its 31 km spatial resolution, will be unlikely effective for the detecting of MCs, as 48% of the MCs could be potentially missed or poorly resolved. In [4,6,9] it is demonstrated that both number of MCs and associated wind speeds in modern reanalyses are significantly underestimated compared to satellite observations of cloud signatures and wind speeds revealed by scatterometers in MCs.

One might argue for the usage of operational analyses for detecting MCs, however these products are influenced by the changing over time model setting, performance of data assimilation system and the volume of assimilated data, thus leading to artificial trends in climatological time scales. Several studies adopted for MCs identification and tracking automated cyclone tracking algorithms originally developed for mid-latitude cyclones [9–12]. These algorithms were applied to the preprocessed (typically hi-pass filtered) reanalysis data and delivered climatological assessments of MC activity in reanalyses. However, reported estimates of MCs numbers, sizes and lifecycle characteristics vary significantly in these studies.

In Zappa et al. [11] demonstrated that ECMWF operational analysis makes it possible to detect up to 70% of the observed PLs, that is much better, than ERA40 and ERA-Interim reanalyses (24% and 45% respectively [9]). Importantly, different hi-pass filters and combinations of criteria used for the post-processing of the MC tracking results may result in 30% spread in the number of PLs [11]. The chosen set of criteria typically represents a compromise between MC definition and data resolution. Laffineur et al. [9] used high-resolution model output (12 km, Meso-NH) with the the threshold on MC size being 500 km, and found the mean diameter of MC to be about 300 km. These results are in agreement with observational studies of [13] and [6], where reported the mean MC diameter of 350 and 300 km respectively. In a number of studies [11,12,14] the upper limit of MC diameter was set to 1000 km, resulting in the mean values between 500 and 800 km. Thus, the level of uncertainty in characteristics of MCs derived with automated tracking algorithms is still high, especially when compared to scheme-to-scheme uncertainties in identification and tracking midlatitude cyclones [15].

Satellite imagery of cloudiness represents another data source for identification and tracking of MCs. These data allow for visual identification of cloud signatures associated of MCs, however

 doi:10.20944/preprints201809.0361.v1

manual procedure requires enormous effort for build long enough dataset. Pioneering work of Wilhemsen [16] used ten years of consecutive synoptic weather maps, coastal observational stations and several satellite images over the Norwegian and Barents Seas to describe local MCs activity. Later in the 1990s the number of instruments and satellite crossovers increased. This provoked many studies [17–23] evaluating characteristics of MCs in different regions of NH and SH. These studies identified of the major MCs generation regions, their dominant migration directions and cloudiness signature types associated with MCs. Increase in the amount of satellite data allowed for the development of the robust regional climatologies of MCs occurrence and characteristics. For the SH Carleton [22] used twice daily cloudiness imagery of the Western Antarctica and classified for the first time four types of cloud signatures associated with PLs (comma, spiral, transitional type, and merry-go-round). This classification has been confirmed later in a many works and is widely used now. Harold et al. [20,21] used daily images for building one of the most detailed dataset of MC characteristics for the Nordic Seas (Greenland, Norwegian, Iceland and Northern Seas). Also Harold et al. [20,21] developed a detailed description of the conventional methodology for the identification and tracking of MCs and PLs using satellite IR imageries.

Importantly, most of studies of MCs activity are regional [13,24–27] and cover relatively short time periods [6] due to very costly and time consuming procedure of visual identification and tracking of MCs. Thus, development of the reliable long-term (multiyear) dataset covering the whole circumpolar Arctic or still remains a challenge.

In the last years machine learning methods were found to be quite effective for the classification of different cloud characteristics such as solar disk state and cloud types. In [28–30] different machine learning techniques was used for recognizing cloud types. Methodologies employed included deep convolutional neural networks (DCNNs [31,32]), k-nearest-neighbor classifier and Support Vector Machine and fully-connected neural networks (FCNNs). Krinitskiy [33] used FCNNs for the detection of solar disk state and reported very high accuracy (96.4%) of his method. Liu et al. [34] applied DCNNs to the fixed-size multichannel images to detect extreme weather events and reported the success score of the detection of 89 to 99%. Huang et al. [35] applied the neural network they term “DeepEddy” to the synthetic aperture radar (SAR) images for detection of ocean meso- and submesoscale eddies. Their results are also characterized by high accuracy exceeding 96% success rate. However Deep Learning methods have never been applied for detecting MCs.

DCNNs are known to demonstrate high skills in classification, pattern recognition, and semantic segmentation, when applied to the the 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 to retrieve e.g. wind speeds (L3 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 scale from the initial data [36–39]. In this sense a trained DCNN closely simulates the visual pattern recognition process naturally used by human operator. There exist several state-of-the-art network architectures such as "AlexNet" [31], "VGG16" and "VGG19" [40], "Inception" of several subversions [41], "Xception" [42] and residual networks [43]. Each of these networks has been trained and tested using a range of datasets including the one that is considered as “reference” for the further image processing, the so-called ImageNet [44]. Continuous development of all DCNNs aims to improve the accuracy of the ImageNet classification. Nowadays the existing architectures demonstrate high accuracy in this benchmark with the error rate from 16% to 2% [45].

Interpreting IR and WV satellite mosaics as images and assuming that a human expert detects MCs on these mosaics on the basis of his visual perception, application of DCNN, thus, closely simulates the visual recognition process and looks promising for the detection of MCs. Liu et al. [34] 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".

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 September 2018 doi:10.20944/preprints201809.0361.v1

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 MCs have hardly observable footprint in geophysical variables of reanalyses.

In this study we apply Deep Learning (DL) technique [46–48] to the satellite IR and WV mosaics distributed by Antarctic Meteorological Research Center [49,50]. This allows for the automated identification of MCs cloud signatures. Our focus here exclusively on the capability of DCNNs to identify MCs from satellite imageries of cloudiness and/or water vapor, rather than on the DCNN-based MC tracking.

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 the 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) [49,50]. The 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 through the MC life cycle [6]. These characteristics were used along with the associated cloudiness patterns of MCs from the initial IR and WV mosaics for training DCNNs.

AMRC ASCI mosaics spatially compose observations from geostationary and polar-orbiting satellites and cover the area to the South of the ~40°S with 3-hourly temporal and 5 km spatial resolution (Fig. 1bc). While the IR channel is widely used for MCs identification [20–22,25,26], we also additionally employ the WV channel imagery which provides a better accuracy over the ice-covered ocean, where the IR images are potentially incorrect.

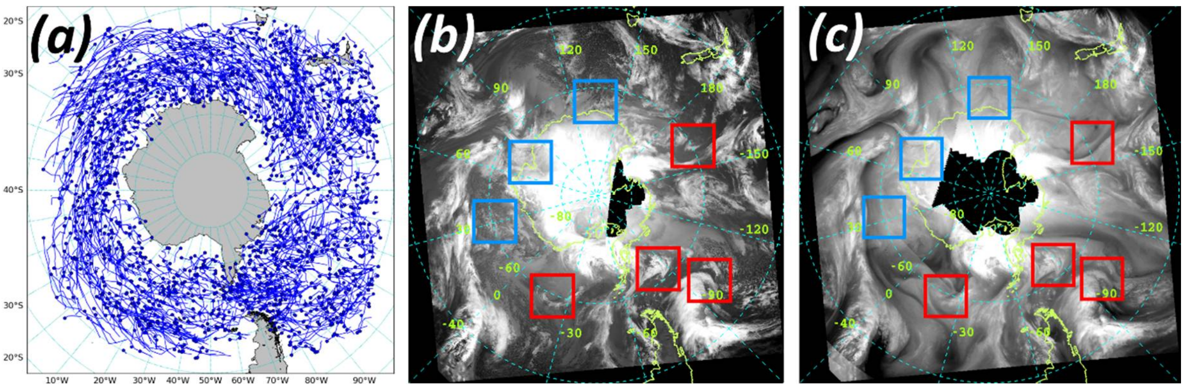

Figure 1. The input for the deep convolutional neural networks (DCNNs). (a) Trajectories of all mesocyclones (MCs) in Southern Ocean MesoCyclones (SOMC) dataset, blue dots mark the point of generation of MC. Snapshots of satellite mosaics for Southern Hemisphere for (b) InfraRed (IR) and (c) Water Vapor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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 September 2018 doi:10.20944/preprints201809.0361.v1

3. Methodology

3.1. Data preprocessing

For training models, we first co-located a square (patch) of 100x100 mosaic pixels (500x500 km) with each MC center location from SOMC dataset (9252 locations in total) (Figure 2a-d). To ensure that (i) each patch covers only one MC and (ii) covers it completely, we require that MC diameter falls into 200-400 km range. Hereafter we call this set of samples 'the true samples'. The chosen set of true samples includes 69% of the whole population of samples in SOMC dataset. We additionally also built the set of 'false samples' for DCNNs training. False samples were generated from the patches that do not consist of MC-associated cloudiness signatures (Figure 2e-h) according to the SOMC dataset. Table 1 summarizes the numbers of true and false samples that both make up source dataset for our further analysis of IR and WV mosaics. The total number of snapshots (both IR and WV) used is 11189 of which 6177 (55%) are the true samples and 5012 (45%) are the false samples (see Fig. 2). In order to unify images in the dataset we normalized them by the maximum and the minimum brightness temperature (in case of IR) over the whole dataset:

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

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

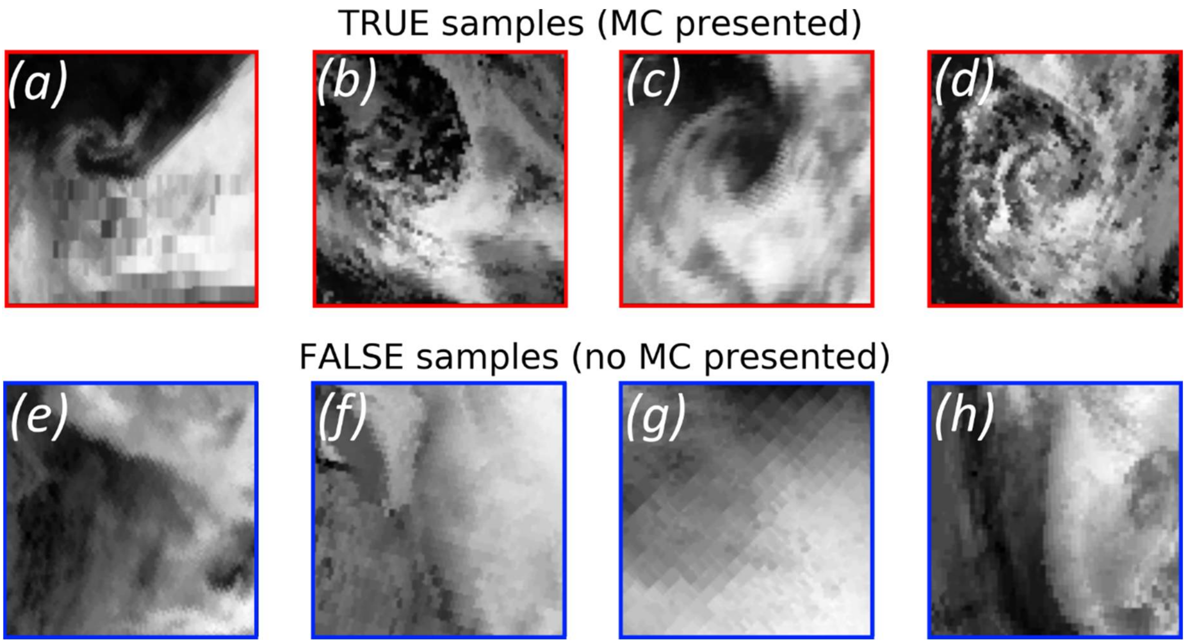

Figure 2. Examples (IR only) of true and false samples for DCNNs training and testing of DCNNs results assessment. 100x100 grid points (500x500km) patches of IR mosaics for (a-d) true samples and false (e-h) samples.

3.2. Formulation of the problem

We consider MC identification as a binary classification problem. As input we use the set of true and false samples (Figure 2), "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 (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 can not 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 highly unbalanced dataset with the true/false ratio being 95/5 it is easy to achieve 95%

accuracy score by just letting the model to repeatedly 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 [51]. While building and training DCNNs for MCs identifications we applied the technique proposed in [36] that 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 direct predicting a target value for the input sample. A very detailed description of this model architecture can be found in [36]. 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 for the input sample.

While handling multiple concurrent and spatially aligned geophysical fields it is important to choose suitable approach. LeCun [36] proposed the DCNN focused on the processing of only grayscale images meaning just one 2D field. In order to handle multiple 2D fields, they may be stacked together to form a 3D matrix by analogy with colorful images which have three color channels: red, green and blue. This approach can be applied when one uses pre-trained networks like AlexNet [31], VGG16 [40], ResNet [43] or similar architectures because of the original purpose of these networks to classify colorful images. However, this approach should be exploited carefully when applied to geophysical fields, because the mentioned networks were trained using massive datasets (e.g. ImageNet) of real photographed scenes, which means specific dependencies laying between channels (red, green and blue) within each image. In contrast to the stacking approach applied in [34] we use separate CNN branch for each channel (IR and WV) to ensure that we are not limiting the overall quality of the whole network (see Fig. 4). In the following we describe in details each DCNN architecture for both cases: IR+WV (Fig. 4) and IR alone (Fig. 3).

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

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

where $\mathcal{T}$ denotes the dataset and $\|\mathcal{T}\|$ is its total samples count; y_i is expert-defined target value (ground truth), $\hat{y}_i$ is the model decision whether the i-th object contain MC.

In addition to the baseline which is the network proposed in [36] we applied a set of additional approaches commonly used to improve the DCNN accuracy and generalization ability (see Appendix A). Particularly we used Transfer Learning (TL) [52–57], Fine Tuning (FT) [58], Dropout (Do) [59] and dataset augmentation (DA) [60]. TL is a technique that allows to use the network of a specific architecture that was trained on a certain set of data, in a problem of a similar kind. It was shown [52–57] that application of TL approach allows to significantly increase classification quality. Specifically we use the VGG16 [40] network pre-trained on ImageNet [44] dataset. FT is a crucial stage for refining models being used with the TL technique applied, to adapt it to specific tasks and datasets [39] (i.e. to the problem of MCs detection). Dropout and dataset augmentation are the approaches applied to suppress the tendency of a DCNN to overfit meaning

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 September 2018 doi:10.20944/preprints201809.0361.v1

the tendency to lose the classification quality evaluated on a never-seen testing data while preserving or improving the classification quality on a training set of data (see Appendix A).

With these techniques applied in various combinations we constructed six DCNN architectures that are summarized in Table 2. All these architectures are built in the common manner: the one- (for IR only) or two-branched (for IR+WV) convolutional core is followed by the FC classifier. If the convolutional core is one-branched, its output is reshaped and resulting vector is input data for the corresponding FC classifier. If the convolutional core is two-branched, then the output of each branch is reshaped to a vector, and the concatenation product of the two vectors is the input data for the corresponding FC classifier. FC classifier includes hidden FC layers whose count varied from 2 to 4. Nodes (artificial neurons) count of FC1 which is the layer following the convolutional core, is randomly chosen from the set {128, 256, 512, 1024}. Each following FC layer size is twice less than preceding one, but not less than 128. The output layer is fully-connected as well and contains one output node. For example, the structure of FC classifier in terms of nodes count of layers might be the following: {512; 256; 128; 1}. All FC layers are alternated with dropout layers (see Appendix A) in order to prevent overfitting of the model. All trainable layers' activation functions are Rectified Linear Unit (ReLU):

$$\sigma_{ReLU}(z) = \max(0; z)\,, \tag{3}$$

except the output layer whose activation function is sigmoid:

$$\sigma_{sigm}(z) = \frac{1}{1+e^{-\theta z}}\,, \tag{4}$$

where θ are layers' trainable parameters.

For each DCNN structure we trained a set of models as described in details in section 3.5. We also applied ensemble averaging (see Appendix A) of a set of models of identical configurations in a manner of averaging probabilities of true class for each object of the dataset. We term these six ensemble-averaged models the "second-order" models. We also applied ensemble averaging per sample of all trained DCNNs trained in this work. We term this model the "third-order" model.

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

$$\mathcal{L} = \sum_{i=0}^{N}(y_i \log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i))\,, \tag{5}$$

where y_i is the expert-defined ground truth for the target value, $\hat{y}_i$ is the estimated probability of the i-th sample to be true, N is samples count of the training set or a training mini-batch. This loss function is minimized in the space of the model weights using the method of backpropagation of error [61] denoted as "backprop training" in Figures 3,4.

3.4. Proposed DCNN architectures

Six DCNNs that we have constructed are able to perform binary classification on satellite mosaics data (IR alone or IR+WV) represented as grayscale 100x100px images:

1. 1. CNN #1. This model is built "from scratch" which means we haven't used any pre-trained networks. CNN #1 is built in the manner proposed in [36]. We varied sizes of convolutional kernels of each convolutional layers from 3x3 to 5x5. We also varied sizes of subsampling layers' receptive fields from 2x2 to 3x3. For each convolutional layers we varied the number of convolutional kernels: 8, 16, 32, 64 and 100. The network convolutional core consists of three convolutional layers alternated with subsampling layers. Each pair of convolutional and subsampling layers is followed by dropout layer. CNN #1 is one-branched and objects are described by IR snapshots only.
2. CNN #2. This model is built "from scratch" with two separate branches - for IR and WV data. Convolutional core of each branch is built in the same manner as convolutional core for CNN #1

 doi:10.20944/preprints201809.0361.v1

and as proposed in [36]. 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 Transfer Learning approach. We used VGG16 pre-trained convolutional core to construct this model. None of VGG16 weights was 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 snapshots only. CNN #3 structure is shown in Fig. 3.
4. CNN #4. This model is two-branched, and each branch of convolutional core is built with Transfer Learning approach, in the same manner as convolutional core of CNN #3. Input data are IR and WV. None of VGG16 weights of this model in any of two branches was 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 Transfer Learning and Fine Tuning approaches. We built 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 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 convolutional core of CNN #5. The last of five blocks of each VGG16 convolutional cores were turned to be trainable. Input data are IR and WV snapshots of dataset samples. CNN #6 structure is shown in Fig. 4.

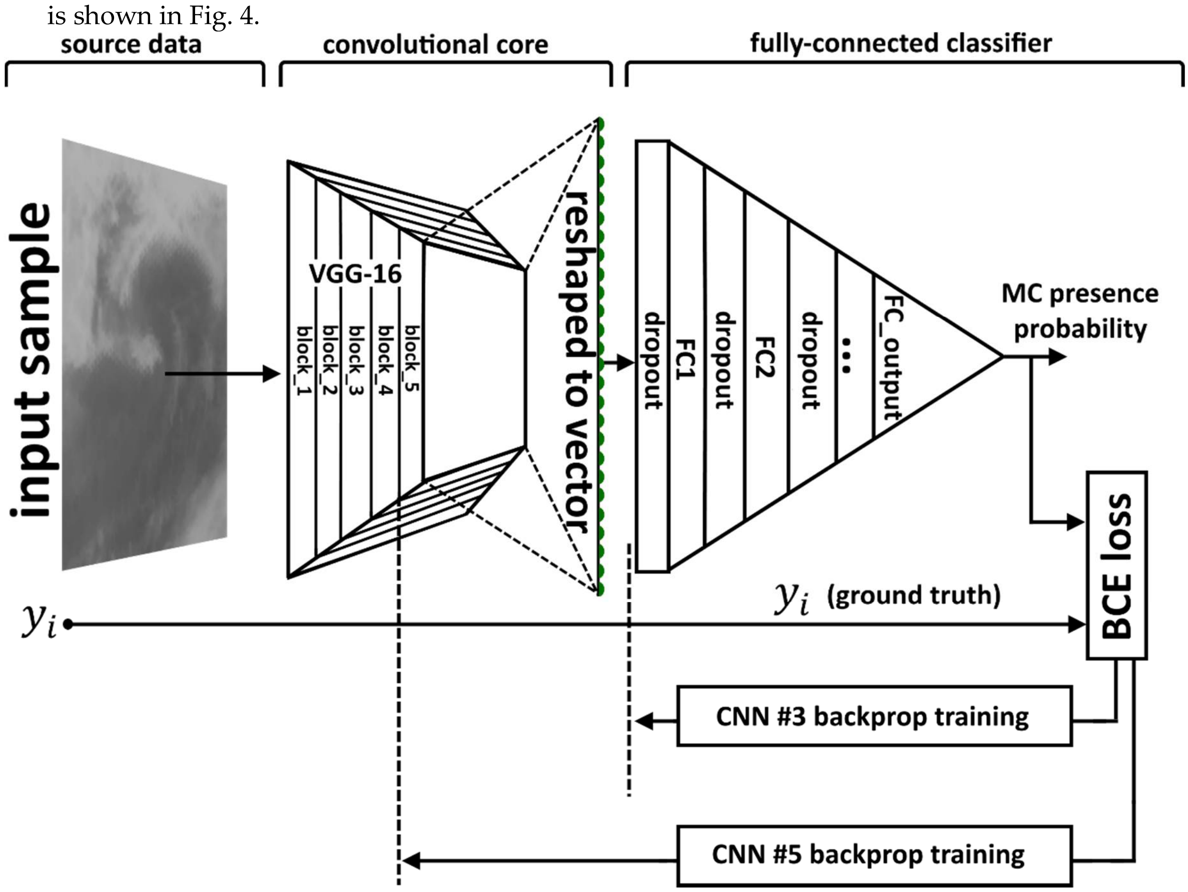

Figure 3. CNN #3 and CNN #5 structures. Green dots denote elements of the convolutional core output reshaped to a vector, which is the fully-connected classifier input data.

 doi:10.20944/preprints201809.0361.v1

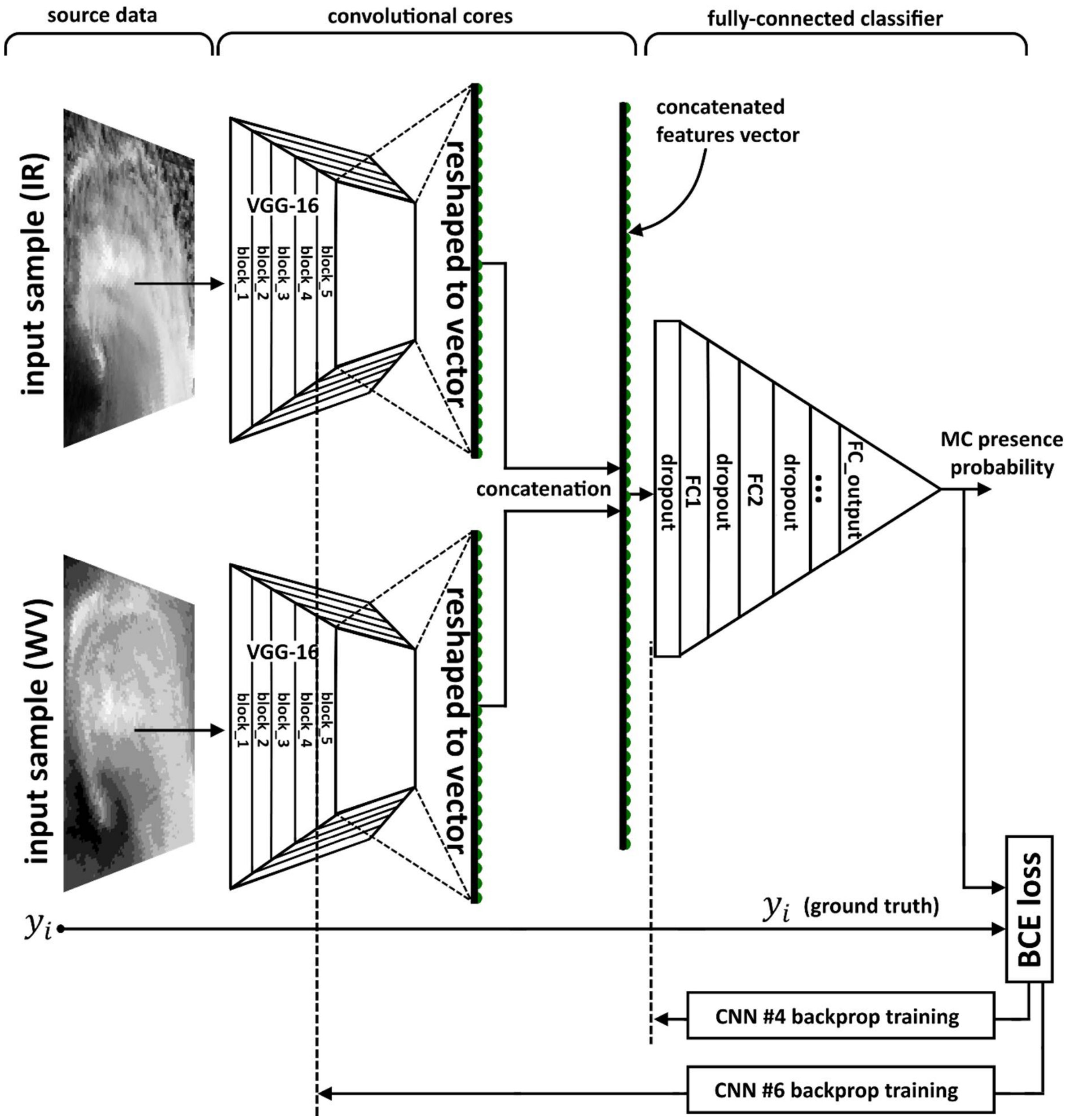

Figure 4. CNN #4 and CNN #6 structures. Green dots denote elements of convolutional cores outputs reshaped to vectors, which are, being concatenated to a combined features vector, the fully-connected classifier input data.

3.5. Computational experiment design

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

- size of FC1 (its nodes number)
- convolutional kernels count for each convolutional layer
- sizes of convolutional kernels
- sizes of receptive fields of subsampling layers

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. We trained several (typically 14-18) models with the best hyper-parameters configuration on the training set for each architecture. Then we drop models with the maximal and minimal accuracy score estimated with the cross-validation approach. The rest of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 September 2018 doi:10.20944/preprints201809.0361.v1

the models are evaluated on the "never-seen by the model" testing set. We estimated the accuracy score for each individual model and also 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 and third-order models (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 on the basis of the Data Center of FEB RAS [62] and Deep Learning computational resources of Sea-Air Interactions Laboratory of 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.

4. Results

The designed DCNNs was applied for the detection of Antarctic MCs for the period from June to September 2004. Summary of the results of 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 the 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 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 [36], 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 presented in Fig. 5 a-d. Confusion matrices and ROC curves for all evaluated models are presented in Appendix C. Figure 5 clearly shows 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 one hand and the remaining models on the other hand shows that DCNNs built with the use of Transfer Learning technique demonstrate better performance compared to the models built "from scratch". Moreover, accuracy score variances of CNN #1 and CNN #2 are higher than for the other architectures. Thus, models built with Transfer Learning approach seem to be more stable, and their generalization ability is better.

 doi:10.20944/preprints201809.0361.v1

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 the model accuracy score. Comparison of models within each pair of the network configurations (CNN #3 vs CNN #5; CNN #4 vs CNN #6) demonstrate that Fine Tuning approach does not provide significant improvement of the accuracy score in case of such a small size of 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 the 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 Fig. 5ef.

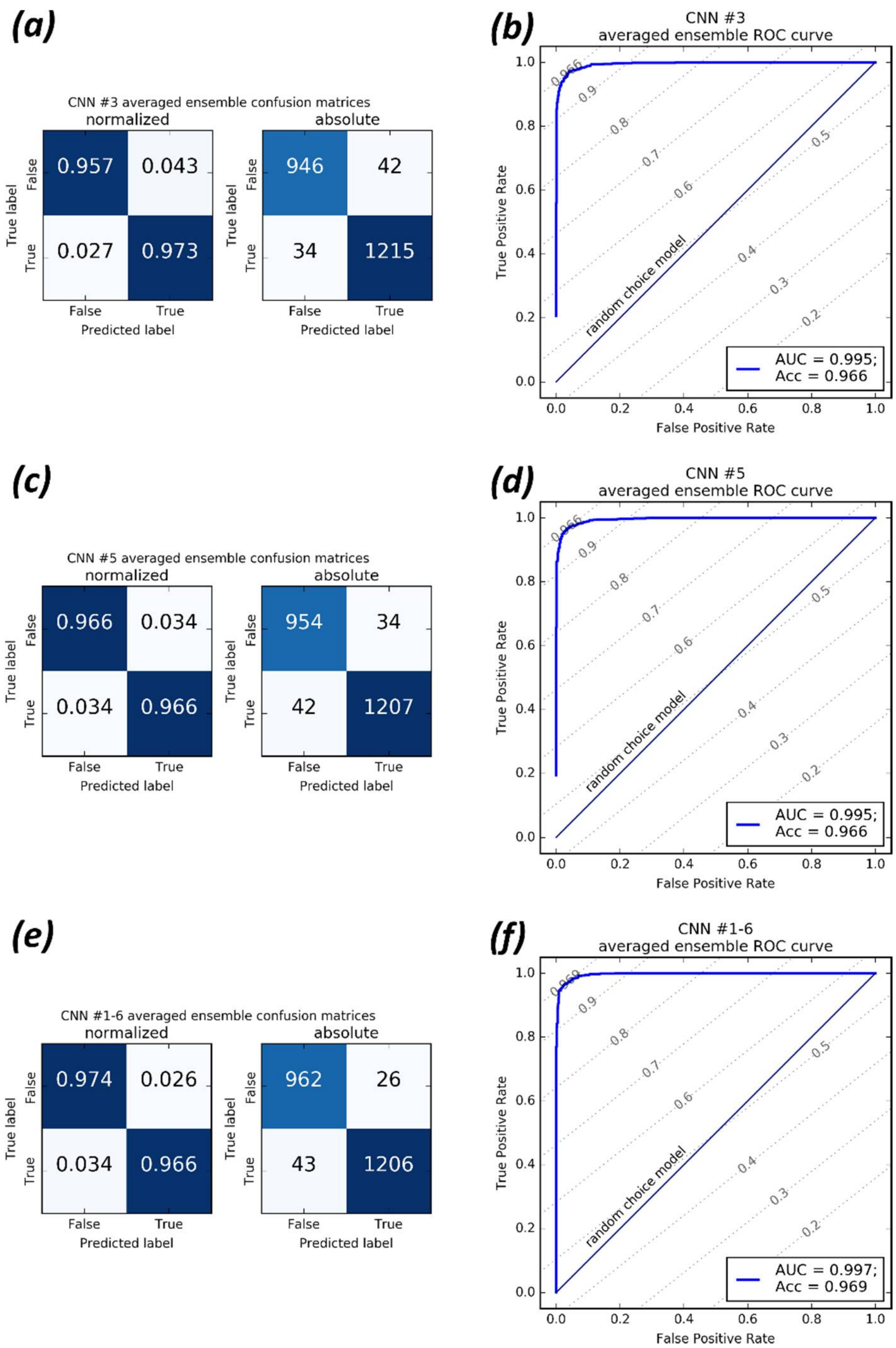

Figure 5. Confusion matrices and receiver operating characteristic curve for (a,b) CNN #3 and (c,d) CNN #5, both with the ensemble averaging approach applied (second-order models); and (e,f) third-order model CNN #1-6 averaged ensemble.

 doi:10.20944/preprints201809.0361.v1

Figure 6 demonstrates four main types of false classified objects. The first and the second types are the ones for which IR data are missing completely or partially. One more type is the one for which the source satellite data were suspected to be corrupted. These three types of classifier errors originating from the lack or corruption of the source data. For the fourth type the source satellite data were realistic but the classifier has done a mistake. Thus some of false classifications are the model mistakes, and some are associated with the labeling issue where human expert could guess on the MC propagation over the area with missing or corrupted satellite data.

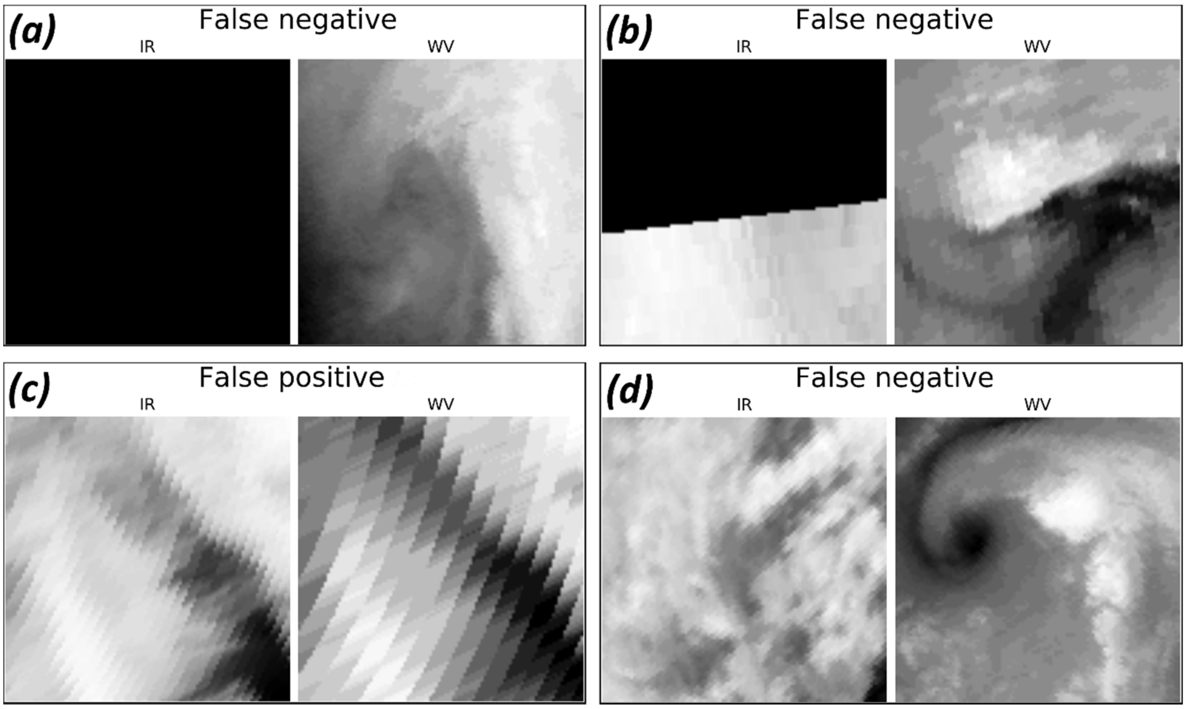

Figure 6. False classified objects.

5. Conclusions and outlook

In this study we present an adaptation of DCNN approach resulted in an algorithm for the detection of MCs from satellite imageries of cloudiness. The DCNN technique shows a very high accuracy in recognition of MCs cloud signatures, with the best accuracy score of 97% is reached by the usage of the third-order ensemble averaging model (6 models ensemble) and combination of both IR and WV images as input. We access the accuracy of MCs identification by comparison of identified MCs (true/false - image contain MC/no MC on the image parameter) with the reference dataset of [6]. We demonstrate that deep convolutional networks are capable for the effective detection of polar mesocyclone signatures in satellite imageries.

It was also shown that the accuracy of MCs detection by DCNNs is sensitive to the single (IR only) or double (IR+WV) input data usage. IR+WV combination provide significant improvement of the detection of MCs and allow a weak DCNN (CNN #2) to detect MCs with higher accuracy compared to the weak CNN #1 (89.3% and 96.3% correspondingly). The computational cost of DCNN training and hyper-parameters optimization for deep neural networks are time- and computational-consuming. However, once trained, the computational cost of the DCNN inference is low. Furthermore, the trained DCNN performs much faster compared to human expert. Another advantage of the proposed method is the low computational cost of data preprocessing that allows to process satellite imageries in real time or to process large amounts of collected satellite data.

We plan to extend the usage of this set of DCNNs (Table 2) for the development of MCs tracking method based on machine learning and satellite IR and WV mosaics. These efforts would be mainly focused onto the development of the optimal choice of the “cut-off” window that has to be applied

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 September 2018 doi:10.20944/preprints201809.0361.v1

to the satellite mosaic. In the case of sliding-window approach (e.g. running the 500x500km sliding window through the mosaics) the virtual testing dataset of the whole mosaic is highly unbalanced, so a model with non-zero FPR evaluated on balanced dataset would produce much higher FPR. In the future, instead of the sliding-window, the Unet-like [63] architecture should be considered with the binary semantic segmentation problem formulation. Considering MC tracking development, an approach proposed in a number of face recognition studies should be reassuring [64,65]. This approach can be applied in a manner of triple-based training of the DCNN to estimate a measure of similarity between one particular MC signatures in consecutive satellite mosaics.

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 [51]. 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 [36]). 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 [36] several works [36–39] 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 [36] 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. In [36] it is suggested to use the neural classifier, and this is now 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 September 2018 doi:10.20944/preprints201809.0361.v1

image). For the integrated water vapor or radio-brightness temperature, $d = 1$. A convolutional layer and subsampling layer are described in details in [36]. 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 [36]). 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 by a 3D matrix. This matrix, being reshaped into a vector, is passed as the input to the FC classifier (see Figures 3,4). The outcome of the the whole model is the probability of each class for the input sample. In the case of binary classification, the FC classifier has one output unit, producing probability of MC presence for the input sample.

In addition to the basic approach proposed in [36] a number of techniques may be applied. Using them one can construct and train DCNNs of various accuracy and various generalization abilities which is characterized by the quality of a model estimated on a never-seen test data.

A.1. Transfer learning

One of the additional approaches is Transfer Learning [52–57]. Generally, this technique focuses on storing the knowledge obtained by some network while being trained for one problem and applying it to another problem of a similar kind. In practice, this approach implies the DCNN structure to be built using some part of a network previously trained on a considerable amount of data, for example, ImageNet [44]. In these terms, VGG16 [40] is not only an efficient architecture, but also the pre-trained network containing optimized weights values (also known as network parameters). Best practice for building a new advanced DCNN based on transfer learning approach is to compose it using convolutional core of the pre-trained model (e.g. VGG16) followed by a new FC neural classifier. Weights of the convolutional part in this case are fixed, and only FC part is optimized. In this approach, the convolutional core may be considered as a feature extractor (see [36]), which computes a highly relevant low-dimensional (compared to original samples dimensionality) vector, representing the data (e.g. “reshaped to vector” output of the convolutional core in Fig. 3).

A.2. Fine Tuning

Transfer Learning approach relies on the similarity of data distributions within two datasets. But in the case of significant differences, for example in terms of Kullback–Leibler divergence between some particular feature approximated probability distributions, the new FC classifier capabilities may not cover all those differences. In this case, some layers of the convolutional core, that are close to FC classifier, can be turned on to be optimized (the so-called Fine Tuning). Regarding DCNNs application to satellite mosaics, we have to consider that VGG16 was optimized on ImageNet dataset which contains everyday-observed objects like buildings, dogs, cats, cars etc., without any satellite imageries or even clouds. So FT approach can be considered as a promising approach when composing MC-detecting DCNN at IR and WV satellite mosaics data.

A.3. Preventing overfitting

Machine learning models and neural networks in particular may vary in terms of complexity. In the case of too strong model there exist an overfitting problem: the effect of poor target prediction quality on unseen data concurrently with nearly exact prediction of target values on training data. There are several state-of-the-art approaches to prevent overfitting of neural networks. We used most fruitful and reliable ones are: dropout [59] and data augmentation also called auxiliary variables [60]. We also used ensemble averaging of models outcome.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 September 2018 doi:10.20944/preprints201809.0361.v1

A.4. Preventing overfitting with dropout

Dropout approach is the way of preventing overfit with a computationally inexpensive but still powerful method of regularizing neural networks through bagging [66] and virtually ensembling models of similar architecture. Bagging involves training multiple models and testing each of them on test samples. Since training and evaluating of deep neural networks tend to be time-consuming and computationally expensive, the original bagging approach [66] seems to be impractical. With the dropout approach applied, the network may be thought as an ensemble of all sub-networks that can be composed by removing non-output nodes from the base network. In practice, this approach is implemented by dropout layer which turns the preceding layer output to zero for each node with some probability p. This procedure repeats for each mini-batch at the training time. At the inference time, the dropout approach involves network weights scaling by $1/p$. Each of our models includes dropout layers between trainable layers. Rate p was set to 0.1 for each dropout layer of each model.

A.5. Preventing overfitting with dataset augmentation

Dataset augmentation is the state-of-the-art way to make a machine learning model generalize better. When available dataset size is limited, the way to get around is to generate fake data which should be similar to real samples. Best practice for DCNNs is generating fake samples adding some noise or applying slight transformations like shift, shear, rotation, scaling etc. Formally, with data augmentation one can increase variability of features of the original dataset and substantially extend its size. This approach often improves generalization ability of the trained model.

We trained each of our models with data augmentation approach applied. The rotation angle range was 90° in both direction; independent width and height scaling performed within range from 0.8 to 1.2; zoom range from 0.8 to 1.2; shear angle range from -2° to 2°. We didn't use flipping upside-down and left-to-right.

A.6. Preventing overfitting with ensemble averaging

In general, during the parameters optimization (learning process) each DCNN converges to a local minimum of the loss function in the space of its weights. The training process starts from a randomly generated point of this space. So due to a non-convexity of loss function, every new DCNN model converges to a new local minimum. Some models may converge to a minimum that is not really close to a global one in terms of loss function value, and thus the quality measure of that model remains poor. Other models may converge to a good minimum that is close to a global one in terms of loss function value, but this proximity may lead to a poor generalization ability which means low quality measure estimated on a testing subset of data. There are approaches for improving the generalization ability of several models that are generally similar, but differ in detailed predictions. In our study we applied simple ensemble averaging [67], which is one of state-of-the-art approaches for improving machine learning models generalization ability. With this approach several models of each architecture are trained, and probabilities of these models are averaged. The prediction of this model is treated as an ensemble outcome:

$$p_i = \frac{\sum_{m=0}^{M} p_i^{(m)}}{M}, \tag{A1}$$

where p_i is the estimated probability of the ensemble of M models for i-th sample to be true; each m-th model`s probability estimation for i-th sample to be true is $p_i^{(m)}$. In this study we applied ensembling on DCNNs of identical architectures. The resulting models we term *second-order models* in this study. They are synthetic ones that are not trained, but are ensembles.

IR+WV snapshots or IR snapshot alone are essentially the object description, and each model that is presented in our study produces the outcome for each object regardless of the description - whether it is IR snapshot alone or IR+WV snapshots. So there is an opportunity to average probability outcomes of all the models of this study. The resulting model that produces averaged probabilities

 doi:10.20944/preprints201809.0361.v1

of the ensemble containing all trained models we term *third-order model*. It is a synthetic one that is not trained, but is an ensemble.

A.7. Adjustment of the probability threshold

The outcome of each model of this study is the estimation of the probability for the sample to be true (i.e. to present an MC). So there is arbitrariness in choosing the threshold of this probability to get the outcome which is binary. The most common way to choose this threshold is the ROC curve analysis. Each point of this curve represents the False Positive Rate (FPR) and True Positive Rate (TPR) combination for the particular probability threshold p_{th} (e.g. see Fig. 5bdf). The model performing true random choice between true and false outcome has a ROC curve on the main diagonal of this plot. The ROC curve of the perfect classifier follows from the point (0.0, 0.0) straight 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 be considered as a measure of model quality. The best model AUC ROC is 1.0, the true random choice model AUC ROC is 0.5, and the worst model AUC ROC is 0.0.

In a range of cases the best accuracy score might not be reached with $p_{th} = 0.5$. The lines of equal accuracy score, as presented in Fig. 5bdf, are diagonal. In case of perfect 50/50 ratio of true/false samples they are parallel to the main diagonal. In case of slight inequality of true and false samples count these lines have slightly different slope as shown in Fig. 5bdf. For each accuracy score there are two, one or no points of the ROC curve intersection with the accuracy isoline. So if a model is represented with a ROC curve, the maximum value of its Acc is located at the point of this curve where the accuracy isoline is tangent to it. For each model of this study including second- and third-order models the optimal probability threshold was estimated based on ROC curve analysis.

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

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

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 [40]; 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

Appendix C. Detailed performance metrics of all DCNN models.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 September 2018 doi:10.20944/preprints201809.0361.v1

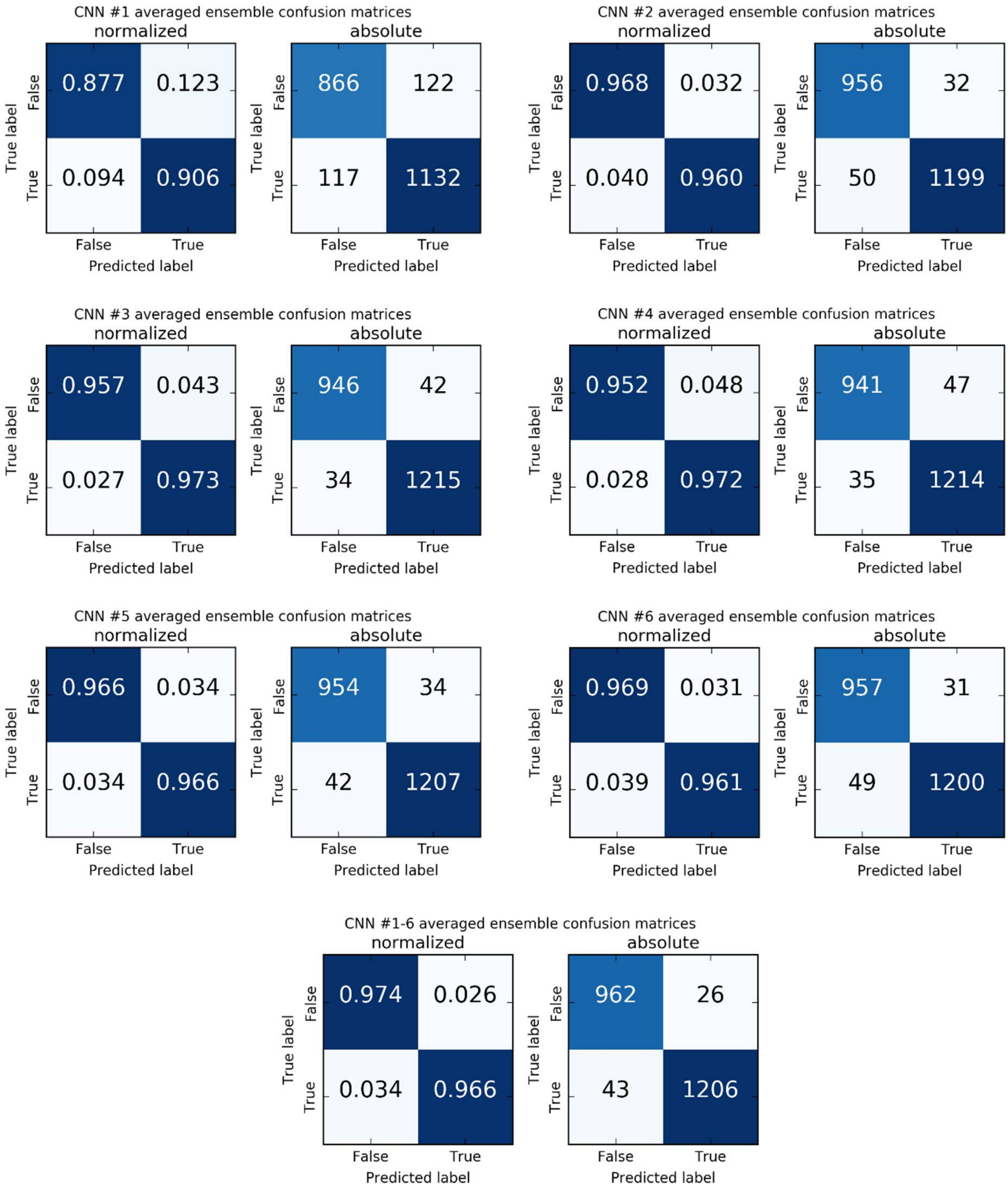

Figure C1. Confusion matrices for all models and the third-order model CNN #1-6 averaged ensemble, computed on test never-seen subset of data. For each architecture the ensemble averaging technique is applied.

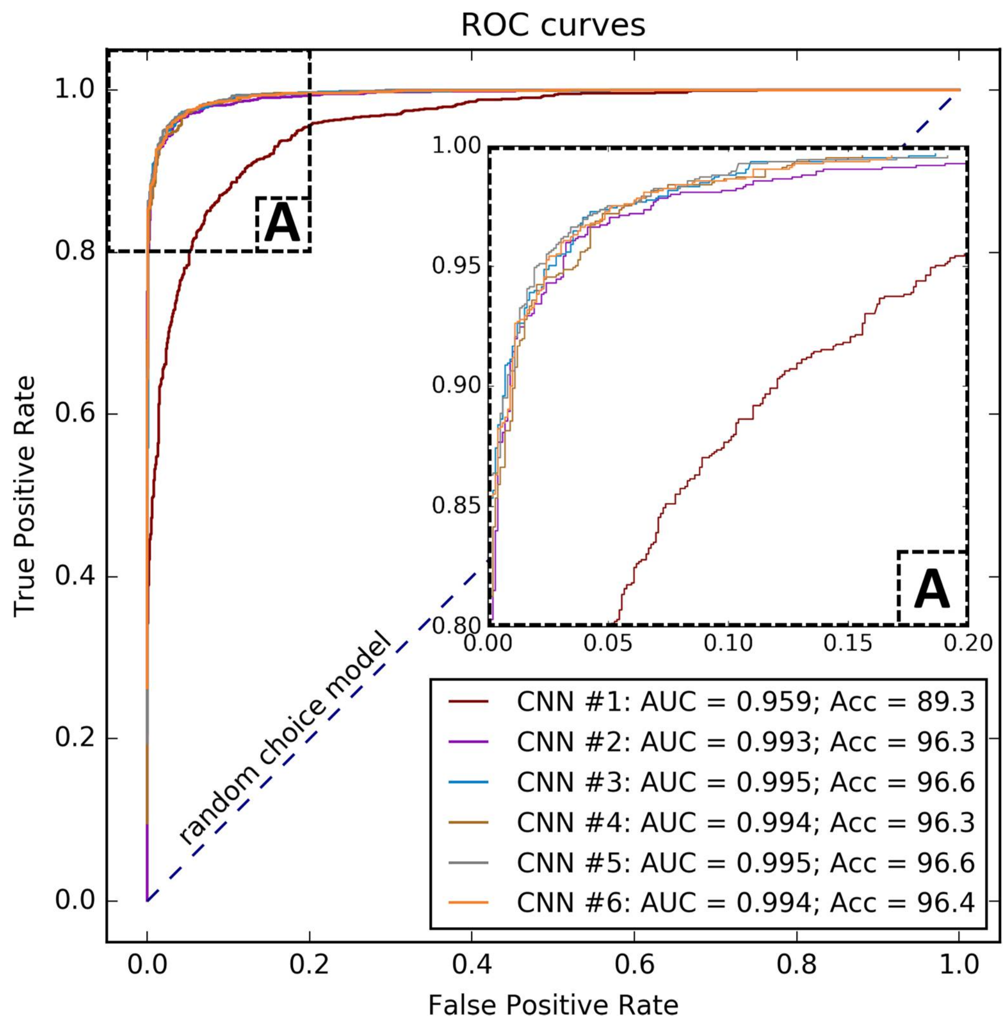

Figure C2. Receiver operating characteristic curves computed on test never-seen subset of data for all models. For each architecture the ensemble averaging technique is applied.

References

1. *Polar Lows: Mesoscale Weather Systems in the Polar Regions*; Rasmussen, E. A., Turner, J., Eds.; Cambridge University Press: Cambridge, 2003; ISBN 978-0-511-52497-4.
2. Marshall, J.; Schott, F. Open-ocean convection: Observations, theory, and models. *Reviews of Geophysics* **1999**, *37*, 1–64, doi:10.1029/98RG02739.
3. Condron, A.; Bigg, G. R.; Renfrew, I. A. Polar Mesoscale Cyclones in the Northeast Atlantic: Comparing Climatologies from ERA-40 and Satellite Imagery. *Mon. Wea. Rev.* **2006**, *134*, 1518–1533, doi:10.1175/MWR3136.1.
4. Condron, A.; Renfrew, I. A. The impact of polar mesoscale storms on northeast Atlantic Ocean circulation. *Nature Geoscience* **2013**, *6*, 34–37, doi:10.1038/ngeo1661.
5. Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical recipes 3rd edition: The art of scientific computing*; Cambridge university press, 2007; ISBN 0-521-88068-8.
6. Verezemskaya, P.; Tilinina, N.; Gulev, S.; Renfrew, I. A.; Lazzara, M. Southern Ocean mesocyclones and polar lows from manually tracked satellite mosaics. *Geophysical Research Letters* **2017**, *44*, 7985–7993, doi:10.1002/2017GL074053.
7. Hersbach, H.; Dee, D. ERA5 reanalysis is in production. *ECMWF newsletter* **2016**, *147*.

8. Barratt, M. ERA5 reanalysis is in production Available online: https://www.ecmwf.int/en/newsletter/147/news/era5-reanalysis-production (accessed on Aug 13, 2018).
9. Laffineur, T.; Claud, C.; Chaboureau, J.-P.; Noer, G. Polar Lows over the Nordic Seas: Improved Representation in ERA-Interim Compared to ERA-40 and the Impact on Downscaled Simulations. *Mon. Wea. Rev.* **2014**, *142*, 2271–2289, doi:10.1175/MWR-D-13-00171.1.
10. Xia, L.; Zahn, M.; Hodges, K.; Feser, F.; Storch, H. A comparison of two identification and tracking methods for polar lows. *Tellus A: Dynamic Meteorology and Oceanography* **2012**, *64*, 17196, doi:10.3402/tellusa.v64i0.17196.
11. Zappa, G.; Shaffrey, L.; Hodges, K. Can Polar Lows be Objectively Identified and Tracked in the ECMWF Operational Analysis and the ERA-Interim Reanalysis? *Mon. Wea. Rev.* **2014**, *142*, 2596–2608, doi:10.1175/MWR-D-14-00064.1.
12. Pezza, A.; Sadler, K.; Uotila, P.; Vihma, T.; Mesquita, M. D. S.; Reid, P. Southern Hemisphere strong polar mesoscale cyclones in high-resolution datasets. *Clim Dyn* **2016**, *47*, 1647–1660, doi:10.1007/s00382-015-2925-2.
13. Rojo, M.; Claud, C.; Mallet, P.-E.; Noer, G.; Carleton, A. M.; Vicomte, M. Polar low tracks over the Nordic Seas: a 14-winter climatic analysis. *Tellus A: Dynamic Meteorology and Oceanography* **2015**, *67*, 24660, doi:10.3402/tellusa.v67.24660.
14. Irving, D.; Simmonds, I.; Keay, K. Mesoscale Cyclone Activity over the Ice-Free Southern Ocean: 1999–2008. *J. Climate* **2010**, *23*, 5404–5420, doi:10.1175/2010JCLI3628.1.
15. Neu, U.; Akperov, M. G.; Bellenbaum, N.; Benestad, R.; Blender, R.; Caballero, R.; Cocozza, A.; Dacre, H. F.; Feng, Y.; Fraedrich, K.; Grieger, J.; Gulev, S.; Hanley, J.; Hewson, T.; Inatsu, M.; Keay, K.; Kew, S. F.; Kindem, I.; Leckebusch, G. C.; Liberato, M. L. R.; Lionello, P.; Mokhov, I. I.; Pinto, J. G.; Raible, C. C.; Reale, M.; Rudeva, I.; Schuster, M.; Simmonds, I.; Sinclair, M.; Sprenger, M.; Tilinina, N. D.; Trigo, I. F.; Ulbrich, S.; Ulbrich, U.; Wang, X. L.; Wernli, H. IMILAST: A Community Effort to Intercompare Extratropical Cyclone Detection and Tracking Algorithms. *Bull. Amer. Meteor. Soc.* **2012**, *94*, 529–547, doi:10.1175/BAMS-D-11-00154.1.
16. Wilhelmsen, K. Climatological study of gale-producing polar lows near Norway. *Tellus A: Dynamic Meteorology and Oceanography* **1985**, *37*, 451–459, doi:10.3402/tellusa.v37i5.11688.
17. Carrasco, J. F.; Bromwich, D. H. Mesoscale cyclogenesis dynamics over the southwestern Ross Sea, Antarctica. *Journal of Geophysical Research: Atmospheres* **1993**, *98*, 12973–12995.
18. Carrasco, J. F.; Bromwich, D. H.; Liu, Z. Mesoscale cyclone activity over Antarctica during 1991: 1. Marie Byrd Land. *Journal of Geophysical Research: Atmospheres* **1997**, *102*, 13923–13937, doi:10.1029/97JD00905.
19. Turner, J.; Thomas, J. P. Summer-season mesoscale cyclones in the bellingshausen-weddell region of the antarctic and links with the synoptic-scale environment. *International Journal of Climatology* **1994**, *14*, 871–894, doi:10.1002/joc.3370140805.
20. Harold, J. M.; Bigg, G. R.; Turner, J. Mesocyclone activity over the North-East Atlantic. Part 1: vortex distribution and variability. *International Journal of Climatology* **1999**, *19*, 1187–1204, doi:10.1002/(SICI)1097-0088(199909)19:11<1187::AID-JOC419>3.0.CO;2-Q.
21. Harold, J. M.; Bigg, G. R.; Turner, J. Mesocyclone activity over the Northeast Atlantic. Part 2: An investigation of causal mechanisms. *International Journal of Climatology* **1999**, *19*, 1283–1299, doi:10.1002/(SICI)1097-0088(199910)19:12<1283::AID-JOC420>3.0.CO;2-T.
22. CARLETON, A. M. On the interpretation and classification of mesoscale cyclones from satellite infrared imagery. *International Journal of Remote Sensing* **1995**, *16*, 2457–2485, doi:10.1080/01431169508954569.

23. Claud, C.; Katsaros, K. B.; Mognard, N. M.; Scott, N. A. Comparative satellite study of mesoscale disturbances in polar regions. *Global Atmos Ocean Syst* **1996**, *4*, 233–273.
24. Claud, C.; Carleton, A. M.; Duchiron, B.; Terray, P. Southern hemisphere winter cold-air mesocyclones: climatic environments and associations with teleconnections. *Climate Dynamics* **2009**, *33*, 383–408, doi:10.1007/s00382-008-0468-5.
25. Blechschmidt, A.-M. A 2-year climatology of polar low events over the Nordic Seas from satellite remote sensing. *Geophysical Research Letters* **2008**, *35*, doi:10.1029/2008GL033706.
26. Noer, G.; Saetra, Ø.; Lien, T.; Gusdal, Y. A climatological study of polar lows in the Nordic Seas. *Quarterly Journal of the Royal Meteorological Society* **2011**, *137*, 1762–1772, doi:10.1002/qj.846.
27. Smirnova, J. E.; Zabolotskikh, E. V.; Bobylev, L. P.; Chapron, B. Statistical characteristics of polar lows over the Nordic Seas based on satellite passive microwave data. *Izv. Atmos. Ocean. Phys.* **2016**, *52*, 1128–1136, doi:10.1134/S0001433816090255.
28. Heinle, A.; Macke, A.; Srivastav, A. Automatic cloud classification of whole sky images. *Atmospheric Measurement Techniques* **2010**, *3*, 557–567, doi:10.5194/amt-3-557-2010.
29. Taravat, A.; Frate, F. D.; Cornaro, C.; Vergari, S. Neural Networks and Support Vector Machine Algorithms for Automatic Cloud Classification of Whole-Sky Ground-Based Images. *IEEE Geoscience and Remote Sensing Letters* **2015**, *12*, 666–670, doi:10.1109/LGRS.2014.2356616.
30. Onishi, R.; Sugiyama, D. Deep Convolutional Neural Network for Cloud Coverage Estimation from Snapshot Camera Images. *SOLA* **2017**, *13*, 235–239, doi:10.2151/sola.2017-043.
31. Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*; 2012; pp. 1097–1105.
32. Shin, H.-C.; Roth, H. R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R. M. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging* **2016**, *35*, 1285–1298, doi:10.1109/TMI.2016.2528162.
33. Krinitskiy, M. A. Application of machine learning methods to the solar disk state detection by all-sky images over the ocean. *Oceanology* **2017**, *57*, 265–269, doi:10.1134/S0001437017020126.
34. Liu, Y.; Racah, E.; Correa, J.; Khosrowshahi, A.; Lavers, D.; Kunkel, K.; Wehner, M.; Collins, W. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv preprint arXiv:1605.01156* **2016**.
35. Huang, D.; Du, Y.; He, Q.; Song, W.; Liotta, A. DeepEddy: A simple deep architecture for mesoscale oceanic eddy detection in SAR images. In *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*; 2017; pp. 673–678.
36. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324, doi:10.1109/5.726791.
37. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444, doi:10.1038/nature14539.
38. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F. E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26, doi:10.1016/j.neucom.2016.12.038.
39. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M. S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48, doi:10.1016/j.neucom.2015.09.116.
40. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]* **2014**.

41. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2015; pp. 1–9.
42. Chollet, F. Xception: Deep learning with depthwise separable convolutions, CoRR abs/1610.02357. *URL http://arxiv. org/abs/1610.02357* **2016**.
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2016; pp. 770–778.
44. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*; Ieee, 2009; pp. 248–255.
45. Eckersley, P.; Nasser, Y. AI Progress Measurement Available online: https://www.eff.org/ai/metrics (accessed on Aug 13, 2018).
46. Deng, L.; Yu, D. Deep Learning: Methods and Applications. *SIG* **2014**, *7*, 197–387, doi:10.1561/2000000039.
47. Deng, L. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing* **2014**, *3*, doi:10.1017/atsip.2013.9.
48. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural networks* **2015**, *61*, 85–117.
49. Lazzara, M. A.; Keller, L. M.; Stearns, C. R.; Thom, J. E.; Weidner, G. A. Antarctic satellite meteorology: applications for weather forecasting. *Monthly Weather Review* **2003**, *131*, 371–383.
50. Kohrs, R. A.; Lazzara, M. A.; Robaidek, J. O.; Santek, D. A.; Knuth, S. L. Global satellite composites — 20 years of evolution. *Atmospheric Research* **2014**, *135–136*, 8–34, doi:10.1016/j.atmosres.2013.07.023.
51. Simard, P. Y.; Steinkraus, D.; Platt, J. C. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of Seventh International Conference on Document Analysis and Recognition*; IEEE: Edinburgh, Scotland, 2003; p. 958.
52. Pratt, L. Y.; Mostow, J.; Kamm, C. A.; Kamm, A. A. Direct Transfer of Learned Information Among Neural Networks. In *AAAI*; 1991; Vol. 91, pp. 584–589.
53. Caruana, R. Learning Many Related Tasks at the Same Time with Backpropagation. *Advances in neural information processing systems* **1995**, 8.
54. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*; ACM, 2008; pp. 160–167.
55. Pan, S. J.; Yang, Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* **2010**, *22*, 1345–1359, doi:10.1109/TKDE.2009.191.
56. Mesnil, G.; Dauphin, Y.; Glorot, X.; Rifai, S.; Bengio, Y.; Goodfellow, I.; Lavoie, E.; Muller, X.; Desjardins, G.; Warde-Farley, D. Unsupervised and transfer learning challenge: a deep learning approach. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop-Volume 27*; JMLR. org, 2011; pp. 97–111.
57. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2014; pp. 1717–1724.
58. Maclin, R.; Shavlik, J. W. Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In *Proceedings of the 1995 International Joint Conference on AI*; Citeseer: Montreal, Quebec, Canada, 1995; pp. 524–531.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 September 2018 doi:10.20944/preprints201809.0361.v1

59. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **2014**, *15*, 1929–1958.
60. Agakov, F. V.; Barber, D. An Auxiliary Variational Method. In *Neural Information Processing*; Lecture Notes in Computer Science; Springer, Berlin, Heidelberg, 2004; pp. 561–566.
61. Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536, doi:10.1038/323533a0.
62. Sorokin, A. A.; Makogonov, S. V.; Korolev, S. P. The Information Infrastructure for Collective Scientific Work in the Far East of Russia. *Sci. Tech. Inf. Proc.* **2017**, *44*, 302–304, doi:10.3103/S0147688217040153.
63. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*; Lecture Notes in Computer Science; Springer, Cham, 2015; pp. 234–241.
64. Parkhi, O. M.; Vedaldi, A.; Zisserman, A. Deep face recognition. In *BMVC*; 2015; Vol. 1, p. 6.
65. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2015; pp. 815–823.
66. Breiman, L. Bagging predictors. *Mach Learn* **1996**, *24*, 123–140, doi:10.1007/BF00058655.
67. Lincoln, W. P.; Skrzypek, J. Synergy of clustering multiple back propagation networks. In *Advances in neural information processing systems*; 1990; pp. 650–657.