Review of egusphere-2025-3622
Title: MCSeg (v1.0): A Deep Learning Framework for Long-Term Large-Scale Mesoscale Convective
Systems Identification and Precipitation Event Analysis
Aurhors: Peng Li, Zhanao Huang, Yongqiang Yu, Xi Wu, Xiaomeng Huang, and Xiaojie Li

The manuscript describes a deep learning (DL)-based methodology to identify Mesoscale Convective Systems (MCS). The methodology accounts for one single algorithm with different approaches for mid- and low-latitude regions. The performances of the algorithm are compared with other machine learning (ML)-based approaches and with a physical-based approach. The analysis shows comparable results between the new developed algorithm and the physical-based approach, while a general outperformance with respect to the other ML approaches.

The paper is well organized and well written. Although I appreciate the work done by the authors with the development of one single algorithm for both medium and low latitudes, I do not see any particular improvement for the scientific community. As I stated below in the specific comment, an ideal case with perfectly working DL algorithm can exactly reproduce the training and test dataset. In the specific case, the only reason to consider the present work as an added value is the computing time. But, I do not see how a simple threshold-based method can be 200 times slower than a DL based method (hours vs less than two minutes).

Reply to Referee comment 1:

- the characteristics of the MCS have to be reported. Not all the readers are familiar with this precipitating structure.

Response: Thank you for your comment. The definition of MCSs, key morphological characteristics, typical life history and its significance in weather and climate have been added at the beginning of the introduction to help non-professional readers understand.

- line 17 and many others: in sentences like this the correct way to report the reference is to put it all in parentheses: (Schumacher and Jonson, 2006).

Response: We have reviewed and corrected the citation format of the references throughout the text to ensure that all citations similar to "(Schumacher and Jonson, 2006)" are placed in parentheses.

- lines 125-126: the use of IMERG final product is only related to the analysis of the precipitation distribution linked to the MCSs carried out in Section 6 and 7? It is not totally clear from the text.

Response: Thank you for your comment. The original text is not clear enough. As we have stated in Section 3.1 "Data information": IMERG precipitation data is only used for the analysis and statistics of precipitation events associated with the identified MCSS.

- lines 137-139: explain better the area coverage threshold. It is not clear whether there must be spatial continuity up to reach 5000 km2 or not. This affect also the interpretation of the results.

When you show, in Figure 6 or 8 for instance, the green and read areas, how these affect the identification of a MCS? I mean, is there a threshold in number of pixels (and consequently in extension area) to say if a given structure is or not a MCS?

Response: Thank you for raising this key question. During the label production stage: The "real" labels we use to train the model are the result of strictly applying physical thresholds. Specifically, a region must meet spatial continuity and its continuous area must exceed 5,000 square kilometers before it can be marked as an MCSs. Therefore, from the very beginning of the model's training, the learning objective has been this "thresholding filtered" MCSs form.
In the model prediction stage: MCSeg is an end-to-end deep learning model. Through learning, the model has internalized the two key physical criteria of "spatial continuity" and "area threshold" into its network parameters. During the training process, the model learned to distinguish between these two types of features by coming into contact with a large number of positive samples (continuous regions with an area greater than 5000 km^2) and negative samples (smaller or discontinuous convective regions). Therefore, when the model makes inferences about a new set of data, what it directly outputs are the regions that it "believes" conform to the physical definition of MCSs.
Explanation of the colors in Figure 6/8: The white area represents the MCSs region correctly recognized by the model, that is, the part where the model predicts MCSs and overlaps with the label region defined by the strict physical threshold. The red and green areas indicate the missed recognition and over-recognition of the model. That is, the area predicted by the model as MCSs, but not covered by strict physical threshold labels. This reveals the systematic differences or uncertainties between the "MCSs features" learned by the model and the original physical threshold definitions. For instance, the model might also identify some convective tissues that are close to but slightly below the strict area threshold and have similar physical structures, or the more diffuse parts of the MCSs edge as MCS.

- lines 146-148: what the reason to increase so much the image size? I can understand that at the edges of your image you can suffer of padding, but in Section 4 you mention you use a 3x3 convolutional kernel.

Response: Thank you for this important technical question regarding the input image size. The enlargement was not primarily due to padding from the 3×3 convolutional kernel, but was a deliberate preprocessing step to address the geographical dimensions of our raw data and to improve the efficiency and coverage of our data sampling during model training.
Our raw satellite brightness temperature data has a global longitudinal coverage of 1715 pixels and a latitudinal coverage of 5143 pixels. To train the model effectively, we divided this global map into smaller, manageable patches. We chose a patch size of 512×512 pixels as it aligns with common practices in deep learning for segmentation tasks and fits well within GPU memory constraints. However, because 1715 is not an integer multiple of 512, simply tiling the longitudinal dimension would result in underutilization of data near the longitudinal boundaries.
To avoid losing valid data, we applied:
Cyclic padding in longitude: Since the Earth's longitude is periodic, we padded the data along the longitudinal axis to make the total width an integer multiple of 512. This allowed us to create complete 512×512 patches without discarding any longitudinal data.

Non-cyclic padding in latitude: As latitude is not periodic, we applied padding at the northernmost and southernmost edges to align the latitudinal dimension with the patch size. These padded pixels were filled with a constant background brightness temperature value of 300 K

- Table 1 and 2: I do not understand the choice of multiplying the coefficient by 10. Also because at lines 259-260 you state that all metrics, except M, have 1 as perfect score.

Response: This is a negligence in our expression. To more clearly display the subtle differences of these assessment indicators, we uniformly multiplied them by 10 for display. We add explanations at the corresponding positions in the main text, emphasizing that the perfect score corresponds to 10, not 1, to avoid misunderstandings.

- lines 309-314: really the application of a threshold on 240 images (8 images per day for 30 days in a month) requires almost 3 hours? Honestly, it is hard to trust on this. On the other hand, this does not imply that your approach cannot be faster than the threshold-based one. In addition, Section 5 describes the comparison between MCSeg and other DL-based methods, but your conclusion mainly focus on the shorter computational time with respect to the threshold-based method. Another weak point is the choice of the 10 DL-based methods. At line 261 you state, “We selected 10 comparative models designed for various segmentation task”. Basically, you are comparing an MCS oriented method with other “generic” methods. In my opinion, the comparison is not fair.

Response: Thank you for raising this reasonable concern about the reported computation time. We understand that the figure may seem unexpected at first. The stated three-hour processing time is the total time needed for our traditional, CPU-based threshold method to generate MCSS labels for one month of global, high-resolution satellite data. This process is not only a simple pixel-level thresholding. It also includes: identifying all connected cold-cloud regions in each image, accurately calculating the real area of each region, and finally selecting only those regions larger than 5,000 km^2 as valid MCSs. All these steps are done in sequence without special optimization, which makes it slow for processing such large and detailed global data. In comparison, after about ten hours of one-time offline training, our proposed MCSeg deep learning model uses GPU acceleration to process approximately 130 global images per minute. This represents a major improvement in speed.

Regarding the choice of models for comparison, we agree that there are currently no publicly available deep learning models made specifically for MCSS recognition to compare with directly. Therefore, we trained several general-purpose image segmentation models on the same dataset as a baseline. Our goal is not to make a perfectly equal comparison, but to show that a model like MCSeg, which is designed with knowledge of MCSs characteristics, can perform better on this specific scientific task than general models.

We are committed to reproducibility. The code for the threshold method is adapted from Huang et al. (their code and data are public). Our MCSeg model code, trained weights, and the dataset we used are also publicly available. We encourage others to test and verify both the processing speeds and model performance independently, as openness is essential to our research.

- lines 323-324: I disagree with this sentence. If your model was perfect, it would be able to exactly reproduce the training (and the test, supposing perfect generalization capabilities) data. This means that you are not able to enhance the threshold-based method but, at most, to equal it (since your dataset is built exploiting the threshold-based method).

Response: Your theoretical understanding is completely correct. The upper limit of a "perfect" DL model in an ideal situation is to reproduce the label (i.e., the result of the threshold method). However, the actual goal of MCSeg is not to reach this theoretical upper limit, but to achieve computational acceleration while maintaining a high precision comparable to that of the threshold method, thereby making large-scale MCSs feature analysis possible.

- the last part of the paper (Figure 9-12 and Section 7) could be considered a bit out of topic with respect to the rest of the paper. The analysis is useful (no doubt), but cannot be considered a consequence of the development of the MCSeg algorithm. In particular, the results shown in Figure 9 highlight negligible differences between MCSeg and threshold-based algorithms. In addition, vertical and horizontal bands (lines 338-339 – I would add curving bands as well) present in Figure 10 have to be fixed. A data post-processing can be applied in order to remove this issue.

Response: Thank you for your insightful feedback. We agree that it is necessary to clarify and strengthen the connection between algorithm development and subsequent scientific analysis. In response, we made extensive revisions to better integrate these sections into the core narrative of the paper.

To make the connection between the chapters of the paper tighter, we merged the last two chapters of the paper, and now the title is " Comprehensive Validation of the MCSeg Algorithm ". We have redefined comparative analysis as a crucial, climate-focused verification step. The main purpose is to demonstrate that our deep learning model MCSeg can generate statistically and climate-consistent MCSS recognition compared with the physical threshold methods established globally. For the vertical/horizontal strip problem. After the MCSS recognition, we processed it with a strip-removing filter. The revised figure in the manuscript now shows a clean one without any stripes.