









Estimating Coronal Mass Ejection Mass and Kinetic Energy by Fusion of Multiple Deep-learning Models

Khalid A. Alobaid^{1,2,3} , Yasser Abdullaah^{1,2} , Jason T. L. Wang^{1,2} , Haimin Wang^{1,4,5} , Shen Fan^{1,2}, Jialiang Li^{1,2}, Huseyin Cavus^{6,7} , and Vasyi Yurchyshyn⁵ 

¹ Institute for Space Weather Sciences, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA; kaa65@njit.edu

² Department of Computer Science, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA

³ College of Applied Computer Sciences, King Saud University, Riyadh 11451, Saudi Arabia

⁴ Center for Solar-Terrestrial Research, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA

⁵ Big Bear Solar Observatory, New Jersey Institute of Technology, 40386 North Shore Lane, Big Bear City, CA 92314, USA

⁶ Department of Physics, Canakkale Onsekiz Mart University, 17110 Canakkale, Turkey

⁷ Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138, USA

Received 2023 September 8; revised 2023 November 6; accepted 2023 November 13; published 2023 November 29

Abstract

Coronal mass ejections (CMEs) are massive solar eruptions, which have a significant impact on Earth. In this paper, we propose a new method, called DeepCME, to estimate two properties of CMEs, namely, CME mass and kinetic energy. Being able to estimate these properties helps better understand CME dynamics. Our study is based on the CME catalog maintained at the Coordinated Data Analysis Workshops Data Center, which contains all CMEs manually identified since 1996 using the Large Angle and Spectrometric Coronagraph (LASCO) on board the Solar and Heliospheric Observatory. We use LASCO C2 data in the period between 1996 January and 2020 December to train, validate, and test DeepCME through 10-fold cross validation. The DeepCME method is a fusion of three deep-learning models, namely ResNet, InceptionNet, and InceptionResNet. Our fusion model extracts features from LASCO C2 images, effectively combining the learning capabilities of the three component models to jointly estimate the mass and kinetic energy of CMEs. Experimental results show that the fusion model yields a mean relative error (MRE) of 0.013 (0.009, respectively) compared to the MRE of 0.019 (0.017, respectively) of the best component model InceptionResNet (InceptionNet, respectively) in estimating the CME mass (kinetic energy, respectively). To our knowledge, this is the first time that deep learning has been used for CME mass and kinetic energy estimations.

Unified Astronomy Thesaurus concepts: [Solar atmosphere \(1477\)](#); [Solar coronal mass ejections \(310\)](#); [Convolutional neural networks \(1938\)](#)

1. Introduction

Coronal mass ejections (CMEs) are massive solar eruptions that release billions of tons of charged particles into space at high speeds (Lin & Forbes 2000; Webb & Howard 2012). These energetic phenomena are of significant importance, as they have the potential to disrupt the Earth's geomagnetic field, resulting in geomagnetic storms that can damage satellites, communication systems, and power grids (Baker et al. 2004). It is crucial to understand and forecast the properties of CMEs to mitigate their potential harmful impact on our technological infrastructure. The study of CMEs has evolved over the years (e.g., Gopalswamy et al. 2005; Schrijver & Siscoe 2012; Pal et al. 2018; Kilpua et al. 2019; Upendran et al. 2020; Martinić et al. 2022). Early work focused on identifying solar features responsible for CMEs, such as magnetic field configurations and the presence of solar flares (Schrijver & Siscoe 2012). Over time, researchers have developed more advanced techniques, including machine learning and artificial intelligence, for CME analysis (e.g., Bobra & Ilonidis 2016; Liu et al. 2018; Wang et al. 2019; Liu et al. 2020; Alobaid et al. 2022; Guastavino et al. 2023). Deep learning, a subfield of machine learning and

artificial intelligence, is now an effective predictive tool in solar physics (Asensio Ramos et al. 2023).

The mass and kinetic energy of CMEs are important characteristics that help scientists understand the dynamics of CMEs (Carley et al. 2012). Determining the mass and kinetic energy of CMEs has been a long-standing topic in heliophysics (Munro et al. 1979; Poland et al. 1981; Carley et al. 2012; de Koning 2017; Na et al. 2021). Traditionally, CME mass is estimated through observations of white-light coronagraphs, which record the brightness of the ejected material as it scatters sunlight (Carley et al. 2012). When these brightness measurements are converted into mass estimates, researchers can calculate the kinetic energy of a CME. For example, Vourlidis et al. (2010) investigated the dependence of the solar cycle on CME mass and kinetic energy over a full solar cycle (1996–2009) using Large Angle and Spectrometric Coronagraph (LASCO) coronagraph data. The authors discovered a sudden reduction in CME mass in mid-2003 and identified a 6 month periodicity in the ejected mass starting from 2003. Carley et al. (2012) utilized STEREO COR1 and COR2 coronagraphs to estimate the mass of a CME on 2008 December 12, revealing that the CME's dynamics were influenced by magnetic forces at heliocentric distances of less than or equal to 7 solar radii and solar wind drag forces at distances more than or equal to 7 solar radii. In another study, Na et al. (2021) presented a method for estimating the mass of halo CMEs using synthetic CMEs. The authors concluded that



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

the halo CME mass might be underestimated when only the observed CME region was considered.

In this paper, we propose DeepCME, which is a fusion of three deep-learning models, to estimate the CME mass and kinetic energy using Solar and Heliospheric Observatory (SOHO) LASCO C2 data. The three deep-learning models are ResNet, InceptionNet, and InceptionResNet. In Section 2, we describe the data used in our study. Besides LASCO C2 images (Brueckner et al. 1995), we also use the CME catalog, which we refer to as the Coordinated Data Analysis Workshops (CDAW) catalog, maintained at the CDAW Data Center (Yashiro et al. 2004; Gopalswamy et al. 2009). Section 3 presents the architecture and configuration details of DeepCME. Section 4 reports the experimental results. Section 5 presents a discussion and concludes the article.

It should be pointed out that our objective is to understand whether machine learning can capture hidden relationships between LASCO C2 observations and CME properties (mass, kinetic energy, occurrence rate, as well as other attributes documented in the CDAW catalog such as angular width, acceleration, etc.). Our experimental results in Section 4 show that the proposed DeepCME model is capable of inferring the relationships between LASCO C2 images and two important CME properties (mass and kinetic energy). These results demonstrate that deep learning could be a useful tool for helping to better understand CME dynamics. We note that the most recent available CME mass and kinetic energy information in the CDAW catalog is from 2020 December. Since 2021 January, this information has been absent. DeepCME could be used to estimate the missing mass and kinetic energy information in the CDAW catalog from 2021 January to the present. Furthermore, the input of the DeepCME tool is obtained from directly observed images, which are available near real-time. Thus, the tool has the potential to contribute to near-real-time CME mass and kinetic energy predictions. Our work presents the first step toward the application of deep-learning models to the estimation of CME attributes. Additional efforts are needed to explore the use of machine learning to predict the other properties of CMEs.

2. Data

We start by collecting 20,084 CME events, spanning 1996 January–2020 December, from the CDAW catalog.⁸ The mass and kinetic energy values of the CME events range from 1.1×10^{10} to 2.0×10^{17} grams and from 2.2×10^{24} to 4.2×10^{33} erg, respectively. Table 1 shows the statistics of the data. For example, the 25th percentile value v in mass represents that 25% of all mass values lie below v and $(100-25)\% = 75\%$ of all mass values lie above v . The wide ranges of values shown in Table 1 present a challenge to a deep-learning model, as they could potentially hinder the model’s ability to learn the underlying patterns effectively. To overcome this issue, we applied a common logarithmic transformation to the values of mass and kinetic energy. This is a widely used technique to normalize data with large variations (Abramenko & Longcope 2005; Yurchyshyn et al. 2005; Vourlidis et al. 2010). Figure 1 shows the distributions of the mass and kinetic energy values after applying the logarithmic transformation.

Table 1
CME Mass and Kinetic Energy Statistics

Statistic	Mass (grams)	Kinetic Energy (erg)
Mean	1.496×10^{15}	4.746×10^{30}
Median	3.500×10^{14}	1.700×10^{29}
Minimum	1.100×10^{10}	2.200×10^{24}
25th Percentile	1.100×10^{14}	3.000×10^{28}
75th Percentile	1.300×10^{15}	1.000×10^{30}
Maximum	2.000×10^{17}	4.200×10^{33}

For each CME event, we downloaded its corresponding LASCO C2 images (Brueckner et al. 1995) from the European Space Agency SOHO Science Archive⁹ utilizing the SunPy library (SunPy Community et al. 2015). These images, with a size of 1024×1024 , provide a comprehensive view of CMEs during their first appearance at 1.5 solar radii in the LASCO C2 field of view, allowing scientists to capture the initial characteristics of the events. To optimize computational efficiency, we resize the images from their original dimension to a size of 256×256 . To make data handling feasible and ensure a representative sample over years, we randomly selected 10% CME events from each year. C2 images with multiple CME events were excluded from the study.

Following Wang et al. (2019), for each selected CME event, we constructed a base-difference image by subtracting its pre-event image from the image in which the CME appears as a full-grown structure. Here, “full-grown” refers to the last LASCO frame when all three parts of the CME (i.e., its core, cavity, and leading-edge; Bellan 2020) are visible within the field of view. A CME event without either the pre-event image or the image in which the CME appears as a full-grown structure was excluded from the study. Construction of this base-difference image allows us to isolate and highlight the changes explicitly associated with the CME event. Figure 2 illustrates how a base-difference image is constructed.

The above process resulted in a set of 1964 base-difference images corresponding to 1964 selected CME events, where each base-difference image uniquely represents a CME event. For each selected CME event and its corresponding base-difference image, we used the common logarithm of its mass and kinetic energy, respectively, as the ground-truth label for the event. We adopt a 10-fold cross validation scheme in which the set of 1964 images is randomly partitioned into 10 subsets or folds of equal size. In the run i , the fold i is used for testing, and the union of the other nine folds is used for training. A total of 10% of the training set is used for validation. There are 10 folds and, therefore, 10 runs. The mean and standard deviation of the predicted mass and kinetic energy values are calculated over the 10 runs and plotted, respectively.

3. Methodology

3.1. Component Models

To extract features from the base-difference images, we employ three deep-learning models: ResNet50 (He et al. 2016), InceptionV3 (Szegedy et al. 2016), and InceptionResNetV2 (Szegedy et al. 2017). The three deep-learning models are among the most widely used convolutional neural networks for computer vision applications. We also experimented with other

⁸ http://cdaw.gsfc.nasa.gov/CME_list/

⁹ <http://ssa.esac.esa.int/>

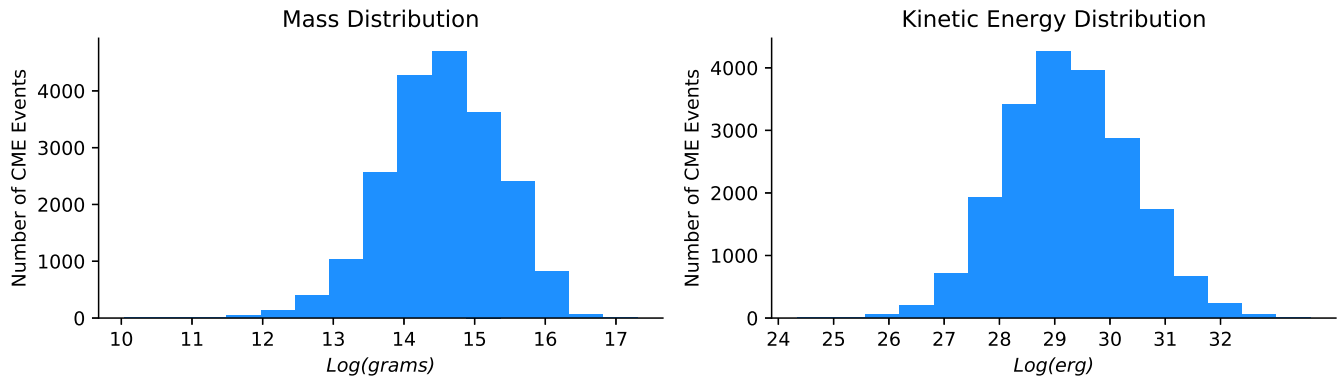


Figure 1. Distributions of the mass and kinetic energy values of the CMEs used in this study.

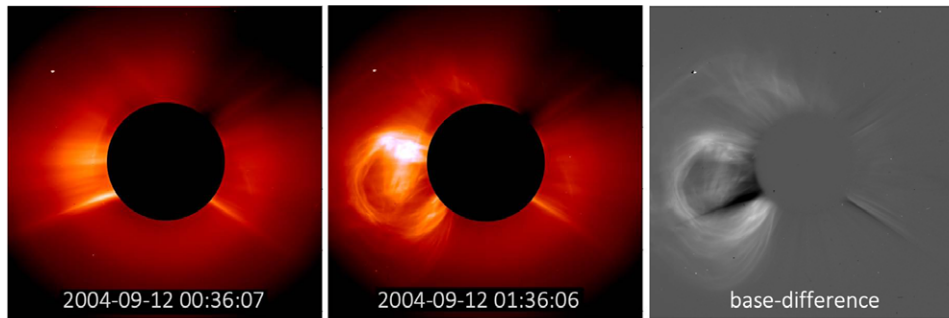


Figure 2. Construction of the base-difference image for the CME event that occurred on 2004 September 12 at 00:36:06 UT. The left panel shows the pre-event image of the CME. The middle panel shows the CME appearing as a full-grown structure. The right panel shows the base-difference image of the CME obtained by subtracting the image in the left panel from the image in the middle panel.

classical models, such as EfficientNet (Tan & Le 2019) and VGGNet (Simonyan & Zisserman 2015), which yielded worse performance.

The ResNet50 model belongs to the class of residual networks (He et al. 2016). It begins with a 7×7 convolutional layer with 64 filters and a stride of 2, followed by a 3×3 max pooling layer with a stride of 2. Next, the model consists of four parts, each containing a sequence of residual blocks. These blocks, also known as bottleneck blocks, are the building blocks of the ResNet50 architecture (He et al. 2016). The InceptionV3 model begins with a 3×3 convolutional layer with 32 filters and a stride of 2, followed by another 3×3 convolutional layer with 32 filters and a stride of 1 (Szegedy et al. 2016). This part is then followed by a 3×3 convolutional layer with 64 filters and a stride of 1, and a 3×3 max pooling layer with a stride of 2. Next, the model contains three inception modules, each with 288 filters, with a grid size of 35×35 . This part is reduced to a 17×17 grid and then to an 8×8 grid (Szegedy et al. 2016). The InceptionResNet model introduces a simple yet effective concept in which it combines the multiscale feature learning of inception modules with the capabilities of ResNet’s residual connections (Szegedy et al. 2017).

The three component models were pretrained on the ImageNet data set (Deng et al. 2009), which contains 1000 object classes with approximately 1.2 million annotated images. To adapt their architectures for the regression tasks of estimating CME mass and kinetic energy, we modify each component model to suit our specific requirements by removing its final fully connected layer and activation function, as the regression tasks require continuous output values instead of discrete class probabilities.

3.2. The Fusion Model

DeepCME is a fusion of the three component models described above. Each input base-difference image, representing a CME event, is fed to the component models, respectively. Each component model is succeeded by a two-dimensional (2D) convolutional layer, followed by five convolutional blocks. The last convolutional block is followed by two dense layers, with 1024 neurons and 1 neuron, respectively. Each component model pipeline predicts an estimated value, respectively. A concatenation layer then takes the median of the three estimated values predicted by the three component model pipelines to produce the final estimated value. Figure 3 shows the architecture of the DeepCME fusion model. Table 2 presents the configuration details of the fusion model.

When estimating the CME mass, we feed all training base-difference images (training CME events) and their corresponding ground-truth labels to DeepCME to train the fusion model. The model is trained for 1000 epochs, with a batch size of 256. We use the adaptive moment estimation optimizer (Adam) and the mean absolute error (MAE) as the loss function (Berk 1992). Table 3 summarizes the hyperparameters used for DeepCME training. During testing, we input each test base-difference image (test CME event) into the trained fusion model, which predicts an estimated CME mass value for the test event. Similarly, when the CME kinetic energy is estimated, we feed all training base-difference images (training CME events) and their corresponding ground-truth labels to DeepCME to train the fusion model. The hyperparameters used in the training are the same as those in Table 3. During testing, we input each test base-difference image (test CME event) into the trained fusion model, which predicts an estimated kinetic energy value for the test event.

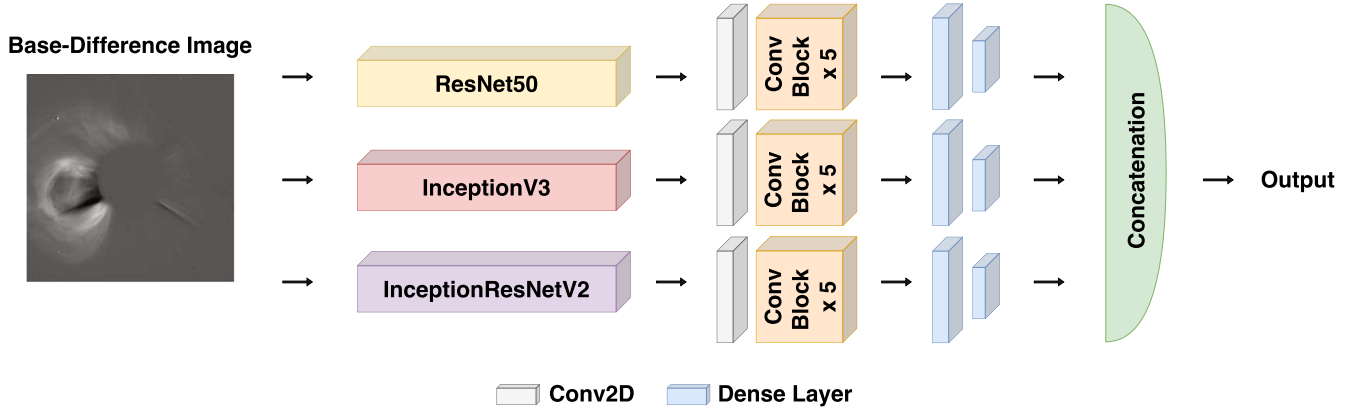


Figure 3. Illustration of the DeepCME architecture. The fusion model begins with three component models, namely ResNet50, InceptionV3, and InceptionResNetV2, each of which is succeeded by a 2D convolutional layer, followed by five convolutional blocks, followed by two dense layers with 1024 neurons and 1 neuron, respectively. The fusion model concludes with a concatenation layer, which produces the output (i.e., the estimated CME mass or kinetic energy value) for the input base-difference image (CME event).

Table 2
Configuration Details of the DeepCME Model

Layer	Type	Number of Filters	Kernel Size	Stride	Regularization	Activation	Output
Conv2D	Convolutional	64	11×11	1	...	LeakyReLU	$8 \times 8 \times 64$
ConvBlock 1	Convolutional	64	11×11	2	Batch Norm	LeakyReLU	$4 \times 4 \times 64$
ConvBlock 2	Convolutional	128	11×11	1	Batch Norm	LeakyReLU	$4 \times 4 \times 128$
ConvBlock 3	Convolutional	128	11×11	2	Batch Norm	LeakyReLU	$2 \times 2 \times 128$
ConvBlock 4	Convolutional	256	11×11	1	Batch Norm	LeakyReLU	$2 \times 2 \times 256$
ConvBlock 5	Convolutional	256	11×11	2	Batch Norm	LeakyReLU	$1 \times 1 \times 256$
Dense	Fully Connected	1024
Dense	Fully Connected	Linear	1

Table 3
Hyperparameters for DeepCME Training

Loss Function	Optimizer	Initial Learning Rate	Batch Size	Epoch
MAE	Adam	0.001	256	1000

4. Results

4.1. Performance Metrics

We use four metrics to evaluate the performance of the DeepCME fusion model and its component models. These metrics include the MAE, the mean relative error (MRE), the coefficient of determination (R -squared or R^2), and Pearson's product-moment correlation coefficient (PPMCC; Pearson 1895; Berk 1992; Jiang et al. 2022). In what follows, y_i denotes the true value of the i th base-difference image (CME event) in the test set, \hat{y}_i denotes the predicted value of the i th base-difference image (CME event) in the test set, n is the total number of base-difference images (CME events) in the test set, and $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ denotes the mean of the true values for all base-difference images (CME events) in the test set.

The first metric is defined as

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \quad (1)$$

which calculates the average absolute difference between the predicted value and the true value (Berk 1992). A smaller MAE signifies a better fit of a model to the data, implying the model's better predictive performance.

The second metric is defined as

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right|, \quad (2)$$

which calculates the average relative difference between the predicted value and the true value. A smaller MRE indicates better model performance.

The third metric is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (3)$$

which measures the strength of the relationship between predicted and true values in the test set. It ranges from $-\infty$ to 1, with a higher value indicating better model performance.

The fourth metric is defined as

$$\text{PPMCC} = \frac{\text{Exp}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}, \quad (4)$$

where X and Y represent the predicted values and true values, respectively; μ_X and μ_Y are the mean of X and Y , respectively; σ_X and σ_Y are the standard deviation of X and Y , respectively; and $\text{Exp}(\cdot)$ stands for the expected value. PPMCC measures the linear correlation between predicted and true values in the test set (Pearson 1895). It ranges from -1 to 1 , with -1 indicating a perfect negative correlation, 1 representing a perfect positive correlation, and 0 meaning that there is no correlation.

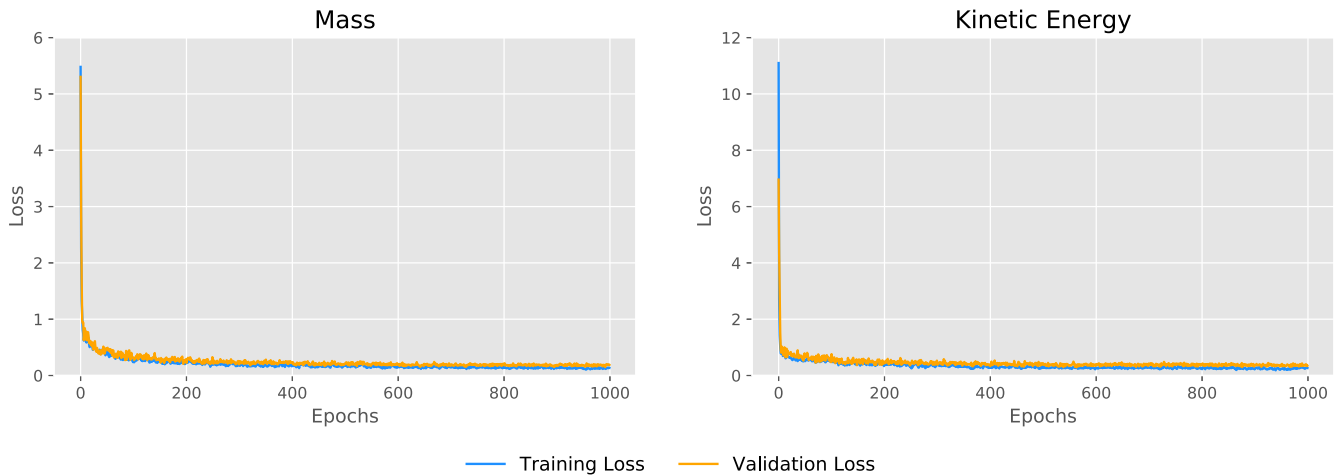


Figure 4. Training and validation learning curves showing DeepCME is a well-fit model in estimating the mass and kinetic energy, respectively, of CMEs.

4.2. Performance Evaluation

We conducted a series of experiments to understand the behavior of DeepCME and evaluate the performance of DeepCME and its three component models (ResNet50, InceptionV3, and InceptionResNetV2). The evaluation was carried out using the 10-fold cross validation scheme described in Section 2, which is a standard technique to detect overfitting. Figure 4 presents the DeepCME training and validation learning curves. The downward and convergence trends in the learning curves demonstrate DeepCME’s ability to learn and generalize well to unseen data, with a decrease in the training loss and validation loss, respectively, as the number of epochs increases. The learning curves in Figure 4 show that DeepCME is a well-fit model.

Figure 5 compares DeepCME with its three component models. In the figure, each colored bar represents the mean of the 10 runs in cross validation, and its associated error bar represents the standard deviation divided by the square root of the number of runs (Alobaid et al. 2022; Iong et al. 2022). When estimating CME mass, the DeepCME model performs better than the other three models, as shown in the left column of Figure 5. DeepCME produces the lowest MAE of 0.190, the lowest MRE of 0.013, the highest R^2 of 0.763, and the highest PPMCC of 0.904. The InceptionV3 model achieves the second-best R^2 of 0.505 and the PPMCC value of 0.791. The InceptionResNetV2 model ranks second in MAE and MRE with 0.271 and 0.019, respectively. When estimating the kinetic energy of CMEs, the DeepCME model also outperforms the other three models, as shown in the right column of Figure 5. DeepCME achieves the lowest MAE of 0.262, the lowest MRE of 0.009, the highest R^2 of 0.828, and the highest PPMCC of 0.920. The InceptionV3 model ranks second on MAE, MRE, and R^2 with 0.534, 0.017, and -0.19 , respectively. ResNet50 is the second-best model in PPMCC with a value of 0.784. Furthermore, DeepCME has the smallest standard deviation and exhibits the most stable behavior among the four models. This happens because DeepCME works by taking the median of the values predicted by the three component model pipelines, resulting in smoother results than the individual component models.

Figure 6 presents scatterplots that visualize the relationship between the predicted values of DeepCME and the actual values when estimating the CME mass and kinetic energy,

respectively. The X -axis denotes the ground truth values, while the Y -axis denotes the predicted values. It can be seen from Figure 6 that the low mass/kinetic energy predictions deviate more than the high mass/kinetic energy predictions. This happens because there are fewer CMEs with low mass/kinetic energy (see Figure 1), and consequently, DeepCME does not acquire enough knowledge during training to make accurate predictions on them. We further conducted a reliability assessment of DeepCME by dividing the test data into reliable test data and unreliable test data (Nicora et al. 2022). When estimating the mass of CMEs, reliable test data contain CMEs whose mass values range from 15 to 17 log(grams), and unreliable test data contain CMEs whose mass values are less than a threshold, θ log(grams), where θ is 14 and 15, respectively. When estimating the kinetic energy of CMEs, reliable test data contain CMEs whose kinetic energy values range from 30 to 33 log(erg), and unreliable test data contain CMEs whose kinetic energy values are less than a threshold, η log(erg), where η is 29 and 30, respectively. Figure 7 compares the PPMCC values obtained by running DeepCME on reliable test data and unreliable test data, respectively. It can be seen in Figure 7 that predictions with lower mass/kinetic energy values are less reliable (with smaller PPMCC values) than predictions with higher mass/kinetic energy values, a finding consistent with the scatterplots presented in Figure 6.

5. Discussion and Conclusion

We present DeepCME, a deep-learning fusion model designed to estimate the mass and kinetic energy of a CME in the CDAW catalog given the LASCO C2 base-difference image that uniquely represents the event. DeepCME combines the strengths of three component models (ResNet, InceptionNet, and InceptionResNet) to extract features from the base-difference images of CME events and to make predictions. Experimental results based on data from 1996 January to 2020 December using a 10-fold cross validation scheme demonstrate the good performance of DeepCME. The fusion model yields a MRE of 0.013 (0.009, respectively) compared to the MRE of 0.019 (0.017, respectively) of the best component model InceptionResNet (InceptionNet, respectively) in estimating the CME mass (kinetic energy, respectively).

We have used LASCO C2 level 0.5 images in our work. In separate experiments, we adopted level 1.0 images to train and

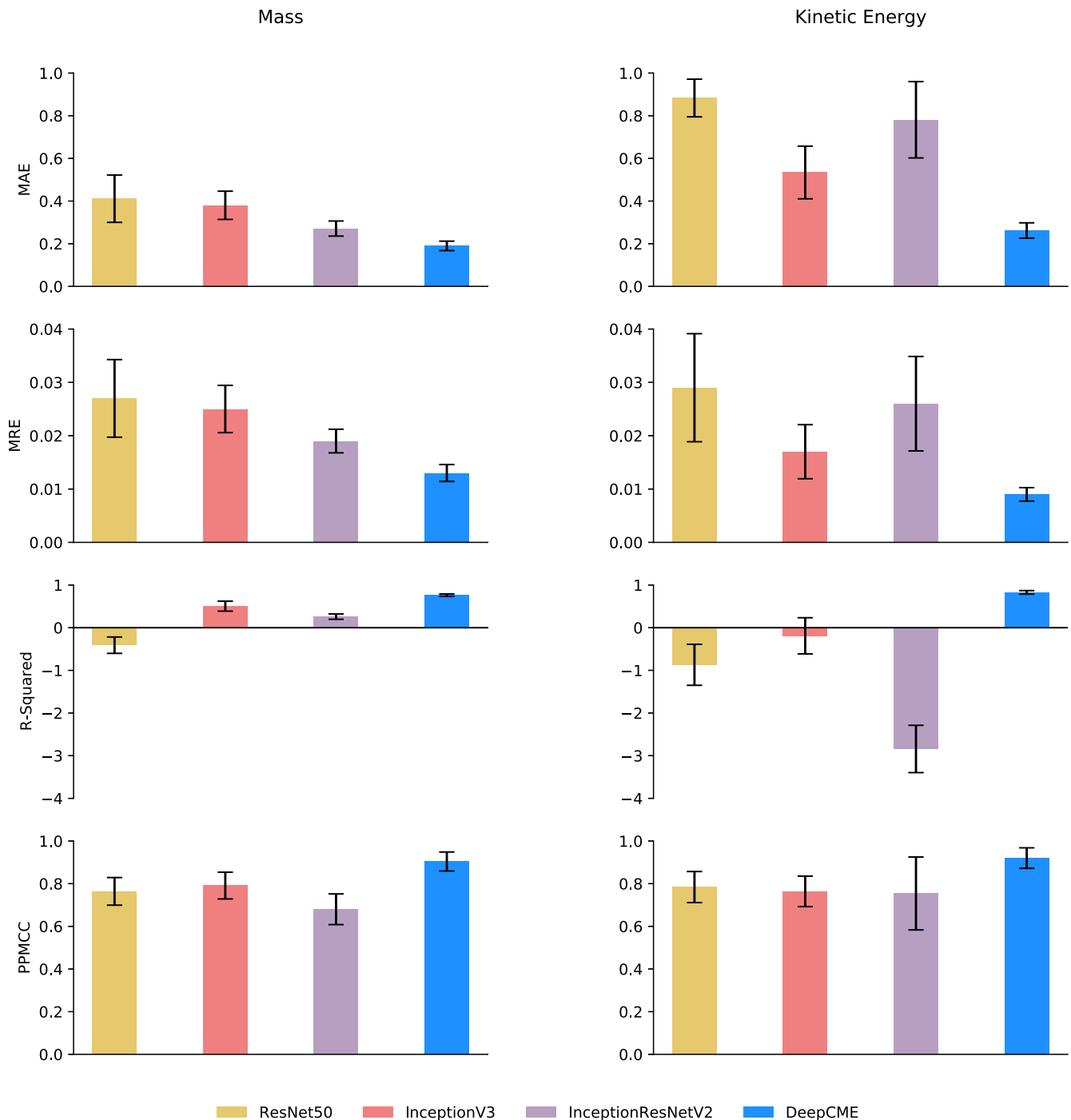


Figure 5. Comparison between DeepCME and its three component models. Left column: performance metric values, displayed by bar charts, obtained by the four models in estimating the mass of CMEs. Right column: performance metric values obtained by the four models in estimating the kinetic energy of CMEs.

test DeepCME. The level 0.5 images are raw data, while the level 1.0 images are calibrated data. Our results show that there is not much difference between the level 0.5 images and the level 1.0 images in terms of prediction accuracy. This happens probably because operations such as image flipping and image warping in the calibration process have no impact on a machine-learning system.

In the study presented here, we used a base-difference image to uniquely represent a CME event. In an additional experiment, we explored an alternative approach in which we used a complete set of LASCO C2 images that spanned a time

frame of 10 minutes before and up to 2 hr after the onset time of a CME to represent the CME event (Wang et al. 2019). All the C2 images shared the same ground-truth label, i.e., the common logarithm of the mass and kinetic energy, respectively, of the event. The results obtained from this experiment indicate that the use of complete sets of images leads to worse performance than the use of unique base-difference images. Specifically, DeepCME yields an MRE of 0.024 (0.021, respectively) when using complete sets of images compared to the MRE of 0.013 (0.009, respectively) obtained by using unique base-difference images in estimating the CME mass

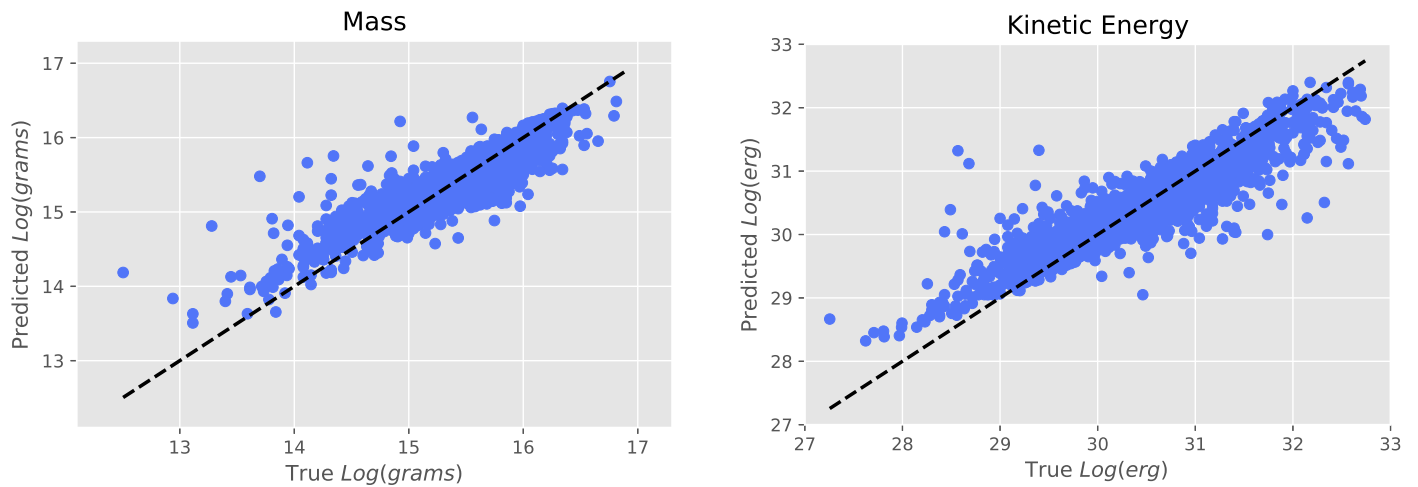


Figure 6. Scatterplots showing DeepCME's predicted values vs. ground truth values in estimating the mass and kinetic energy, respectively, of CMEs.

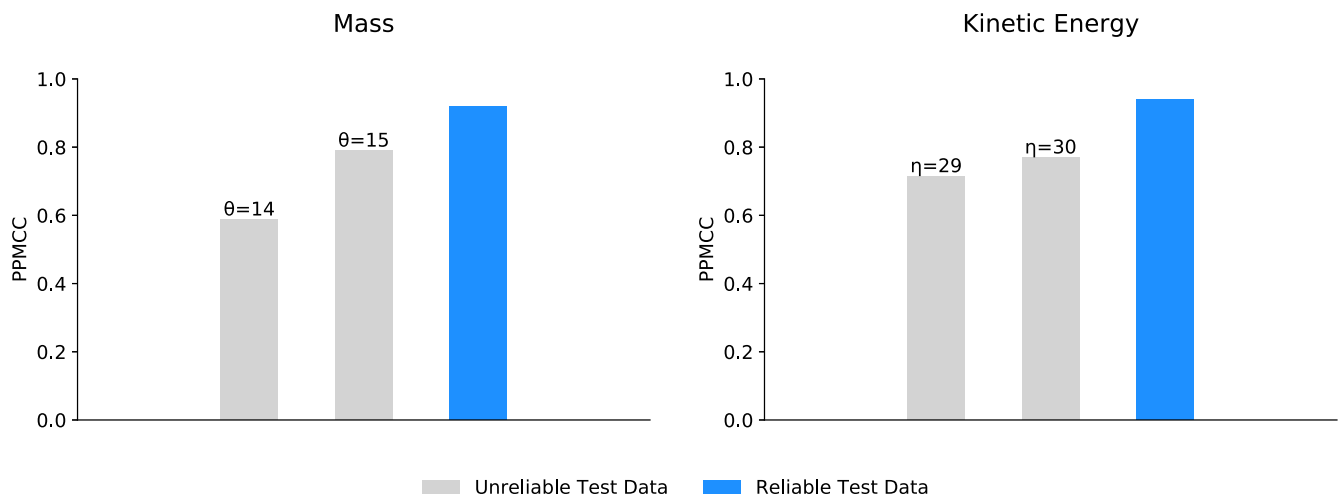


Figure 7. Reliability assessment of DeepCME showing the performance of the model on reliable test data and unreliable test data, respectively.

(kinetic energy, respectively). In theory, one would need to label the different images of a CME event with different kinetic energy values while taking into account the velocity of the CME. However, the CDAW catalog provides only one kinetic energy value for each CME event, rather than one kinetic energy value for each image. Assigning the same ground-truth label to different images of a CME event would confuse a machine-learning model, which would yield worse performance. We conclude that the proposed approach of using unique base-difference images is a viable one for CME mass and kinetic energy estimations.

Acknowledgments

We appreciate the anonymous referee for constructive comments and suggestions. We thank members of the Institute for Space Weather Sciences for fruitful discussions. K.A. is supported by King Saud University, Saudi Arabia. J.W. and H. W. acknowledge support from NSF grants AGS-1927578, AGS-2149748, AGS-2228996, and OAC-2320147. H.C. is supported by the Fulbright Visiting Scholar Program of the Turkish Fulbright Commission. V.Y. is supported by the NSF grant AGS-2300341. The CME catalog used in this work was created and maintained at the CDAW Data Center by NASA and the Catholic University of America in cooperation with the

Naval Research Laboratory. SOHO is an international cooperation project between ESA and NASA.

ORCID iDs

Khalid A. Alobaid  <https://orcid.org/0009-0007-4731-2772>
 Yasser Abdullallah  <https://orcid.org/0000-0003-0792-2270>
 Jason T. L. Wang  <https://orcid.org/0000-0002-2486-1097>
 Haimin Wang  <https://orcid.org/0000-0002-5233-565X>
 Huseyin Cavus  <https://orcid.org/0000-0003-4224-7039>
 Vasyly Yurchyshyn  <https://orcid.org/0000-0001-9982-2175>

References

- Abramenko, V. I., & Longcope, D. W. 2005, *ApJ*, **619**, 1160
 Alobaid, K. A., Abdullallah, Y., Wang, J. T. L., et al. 2022, *FrASS*, **9**, 1013345
 Asensio Ramos, A., Cheung, M. C. M., Chifu, I., & Gafeira, R. 2023, *LRSP*, **20**, 4
 Baker, D. N., Daly, E., Daglis, I., Kappenman, J. G., & Panasyuk, M. 2004, *SpWea*, **2**, S02004
 Bellan, P. M. 2020, *JGRA*, **125**, e28139
 Berk, K. N. 1992, *SIAMR*, **34**, 157
 Bobra, M. G., & Ilonidis, S. 2016, *ApJ*, **821**, 127
 Brueckner, G. E., Howard, R. A., Koomen, M. J., et al. 1995, *SoPh*, **162**, 357
 Carley, E. P., McAteer, R. T. J., & Gallagher, P. T. 2012, *ApJ*, **752**, 36
 de Koning, C. A. 2017, *ApJ*, **844**, 61
 Deng, J., Dong, W., Socher, R., et al. 2009, in 2009 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2009), Proc. of IEEE Conf. on

- Computer Vision and Pattern Recognition (Miami, FL: IEEE Computer Society), 248
- Gopalswamy, N., Aguilar-Rodriguez, E., Yashiro, S., et al. 2005, *JGRA*, **110**, [A12S07](#)
- Gopalswamy, N., Yashiro, S., Michalek, G., et al. 2009, *EM&P*, **104**, [295](#)
- Guastavino, S., Candiani, V., Bemporad, A., et al. 2023, *ApJ*, **954**, [151](#)
- He, K., Zhang, X., Ren, S., & Sun, J. 2016, in 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2016), Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (Las Vegas, NV: IEEE Computer Society), 770
- long, D., Chen, Y., Toth, G., et al. 2022, *SpWea*, **20**, [e2021SW002928](#)
- Jiang, H., Li, Q., Xu, Y., et al. 2022, *ApJ*, **939**, [66](#)
- Kilpua, E. K. J., Lugaz, N., Mays, M. L., & Temmer, M. 2019, *SpWea*, **17**, [498](#)
- Lin, J., & Forbes, T. G. 2000, *JGR*, **105**, [2375](#)
- Liu, H., Liu, C., Wang, J. T. L., & Wang, H. 2020, *ApJ*, **890**, [12](#)
- Liu, J., Ye, Y., Shen, C., Wang, Y., & Erdélyi, R. 2018, *ApJ*, **855**, [109](#)
- Martinić, K., Dumbović, M., Temmer, M., Veronig, A., & Vršnak, B. 2022, *A&A*, **661**, [A155](#)
- Munro, R. H., Gosling, J. T., Hildner, E., et al. 1979, *SoPh*, **61**, [201](#)
- Na, H., Moon, Y.-J., Lee, J.-Y., & Cho, I.-H. 2021, *ApJ*, **906**, [46](#)
- Nicora, G., Rios, M., Abu-Hanna, A., & Bellazzi, R. 2022, *J. Biomed. Informatics*, **127**, [103996](#)
- Pal, S., Nandy, D., Srivastava, N., Gopalswamy, N., & Panda, S. 2018, *ApJ*, **865**, [4](#)
- Pearson, K. 1895, *RSPS*, **58**, [240](#)
- Poland, A. I., Howard, R. A., Koomen, M. J., Michels, D. J., & Sheeley, N. R. J. 1981, *SoPh*, **69**, [169](#)
- Schrijver, C. J., & Siscoe, G. L. 2012, *Heliophysics: Space Storms and Radiation: Causes and Effects* (Cambridge: Cambridge Univ. Press)
- Simonyan, K., & Zisserman, A. 2015, [arXiv:1409.1556](#)
- SunPy Community, T., Mumford, S. J., Christe, S., et al. 2015, *CS&D*, **8**, [014009](#)
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. 2017, in Proc. of the AAAI Conf. on Artificial Intelligence, 31 (San Francisco, CA: AAAI Press), 4278
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016, in Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (Los Alamitos, CA: IEEE Computer Society), 2818
- Tan, M., & Le, Q. V. 2019, in 36th Int. Conf. on Machine Learning (ICML 2019), Proc. of the 36th Int. Conf. on Machine Learning, ed. K. Chaudhuri & R. Salakhutdinov (Long Beach, CA: PMLR), 6105, [http://proceedings.mlr.press/v97/tan19a.html](#)
- Upendran, V., Cheung, M. C. M., Hanasoge, S., & Krishnamurthi, G. 2020, *SpWea*, **18**, [e02478](#)
- Vourlidis, A., Howard, R. A., Esfandiari, E., et al. 2010, *ApJ*, **722**, [1522](#)
- Wang, Y., Liu, J., Jiang, Y., & Erdélyi, R. 2019, *ApJ*, **881**, [15](#)
- Webb, D. F., & Howard, T. A. 2012, *LRSP*, **9**, [3](#)
- Yashiro, S., Gopalswamy, N., Michalek, G., et al. 2004, *JGRA*, **109**, [A07105](#)
- Yurchyshyn, V., Yashiro, S., Abramenko, V., Wang, H., & Gopalswamy, N. 2005, *ApJ*, **619**, [599](#)