AI in Solar Observations: Classifying Active Sunspot Regions

Author: Eduard Plic

KEYWORDS: sunspots; artificial intelligence; convolutional neural networks; McIntosh classification; sunspot drawing

1 INTRODUCTION

The primary motivation of this work is to explore the application of artificial intelligence (AI) in solar astronomy, specifically in the classification of sunspots. While the term "artificial intelligence" has become widely known due to the rise of chatbots and image-generating tools, it is often misunderstood and misused.

As AI continues to gain traction across various scientific disciplines, this study investigates its potential role in solar research by developing a convolutional neural network (CNN) model designed to classify individual sunspot groups according to the McIntosh classification system. The model is trained using hand-drawn solar observations, which provide a unique historical dataset that remains valuable for scientific analysis.

2 SUNSPOTS

Sunspots are dark regions on the Sun's photosphere, first documented in ancient Asia and studied more closely after the invention of the telescope in the 17th century. They are caused by magnetic field distortions associated with the 11-year solar cycle. Each sunspot consists of a dark central region called the umbra (with a temperature of approximately 4,000 K), surrounded by a lighter, filamented area known as the penumbra. These spots are also observable in the visible spectrum, as shown in Figure 1. Variability of sunspots in size, shape, and spatial distribution provides important data that helps in analyzing solar activity and forecasting related phenomena such as eruptions or auroras.

2.1 Groups of Sunspots

Paradoxically, a sunspot group can even be just a single spot, as it represents a closed magnetic system. Groups may range from one to dozens of spots, with their size, structure, and spot count influencing their lifespan, which can vary from a few hours to several weeks. In Figure 1 (left), approximately four groups are visible.

2.2 Classification of Active Regions

To better describe and understand solar activity, sunspot groups, or active regions, can be categorized by multiple classification systems based on size, polarity, distribution, and more. Currently, there is no adequate method for accurately predicting the occurrence of potentially dangerous eruptions, as this remains a very complex task.

2.2.1 McIntosh Classification

The McIntosh classification, developed in 1966, consists of three nearly independent categories that together form a three-letter designation for each sunspot group. For example, groups can be labeled as Axx, Dai, or Eso. The first letter (A, B, C, D, E, F, H) refers to the overall size of the group. The second letter (x, r, s, a, h, k) characterizes the penumbra of the largest spot in the group. The third letter (x, i, o, c) describes the distribution of spots within the group. For examples and detailed descriptions of the classes, see Figures 2 and 3.

2.3 Sunspot Drawing

Sunspot drawing, visible on Figure 4, first used by Galileo Galilei, involves projecting the Sun onto paper through a telescope and sketching the observed features, as shown in Figure 5. This practice represents a continuous observational record dating back to the 17th century, although only a few Czech observatories still engage in it regularly. The Astronomical Institute is one such institution, offering many of its drawings for free download. In Ondřejov, home to one of the Institute's observatories, solar observation has a long tradition. The first publicly available drawing from there dates back to 1944, and their archive includes data captured during World War II on photographic plates. While some may argue that solar drawing is unnecessary in today’s digital age, it still provides valuable insights. Comparing both types of data helps us better understand periods when only drawings were available.

2.3.1 Information in the Record

The record must include various information: the date and time of the drawing, the observation location, the observer's name, observing conditions, and the drawing number. Furthermore, the record should contain data about the solar disk, including the heliographic latitude and longitude of the solar disk center, the angle of the Sun’s rotational axis tilt, and the Carrington rotation number. The drawn groups of spots should be outlined in rectangles, numbered, and possibly classified. The record should also include information about solar activity, such as the number of facular fields, the Wolf number, the number of spots and groups on the disk, and their distribution across three sectors: central, northern, and southern. Additionally, the record should specify the location of each group, namely their position relative to the center of the drawing. Heliographic coordinates of the group can be then calculated using the following formulas:

$$\rho = \sin^{-1}\left(\frac{vz}{R}\right), \tag{3}$$

$$b = \sin^{-1}(\sin B_0 \cos\rho + \cos B_0 \sin\rho \cos(P - Q)), \tag{4}$$

$$l = \left(\sin^{-1}\frac{\sin\rho \sin(P-Q)}{\cos b}\right) + L_0, \tag{5}$$

where variables are shown and explained on Figure 6. The letters b and l are the sought heliographic coordinates, latitude and longitude, of the group.

3 MACHINE LEARNING

Machine learning refers to methods and algorithms used to perform tasks such as categorization, prediction, similarity detection, or content generation. It powers applications ranging from smart car vision to stock forecasting, as well as tools like ChatGPT and DALL-E. A common approach to training a model is supervised learning, where input data is paired with correct outputs. The model learns patterns from the data, such as recognizing cars and dogs in images, and improves based on feedback.

3.1 Neural Networks

Neural networks consist of neurons that activate based on values received from previous neurons through simple mathematical operations. Each neuron connects to all neurons in the previous layer, with varying connection weights w_i. The intermediate value of a neuron, denoted as z, is the weighted sum of the previous neuron values plus a threshold or bias b, unique to each neuron and independent of the input data:

$$z = \sum_{i=1}^{n}(x_i w_i) + b, \tag{6}$$

After that, an activation function $f(z)$ is applied to add nonlinearity to the learning process, significantly improving the model's training capabilities. The entire process is illustrated in Figure 7 (left). The resulting value of the neuron y can then be formulated as:

$$y = f(z) = f(\sum_{i=1}^{n}(x_i w_i) + b). \tag{7}$$

As the previous text suggests, neurons in neural networks are arranged in layers as seen in Figure 7 (right). The first, called the input layer, simply takes data from the input. It is followed by several hidden layers, each potentially containing a different number of neurons. The last layer is called the output layer. In classification tasks, it has as many neurons as there are output classes, and the values of these neurons represent the probabilities of each class.

For classification models, results can also be interpreted using a confusion matrix. Its rows represent the correct classifications, while columns indicate the model's predictions. Referring to Figure 8, the model predicted classification into four classes: the first class was predicted perfectly; the second with minor errors; samples of the third class were often misclassified as belonging to all classes; and the fourth class was never correctly predicted, most often confused with the second class.

3.1.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a neural network specialized for image processing. Unlike basic feedforward networks, CNNs utilize the spatial layout of 2D input data (pixels) to capture relationships within images. Instead of connecting every neuron to all neurons in the previous layer, CNN neurons focus on smaller regions, enabling operations like max-pooling for better feature extraction. Figure 9 illustrates how the value of a neuron is calculated.

4 SUNSPOT CLASSIFICATION USING AI

The aim of this work was to create a convolutional model capable of predicting the classification of sunspot groups according to the McIntosh system, using data collected so far. We utilized 7,170 solar drawings from the period between 1971 and October 6, 2023. The observatory also provided us with electronic record containing the locations and classifications of 56,375 sunspots, spanning from March 25, 1944, to October 19, 2023. Another reason for choosing the McIntosh classification is that it consists of three nearly independent subclasses, allowing us to evaluate the model's performance for each subclass separately. Furthermore, it is currently the most accurate and widely used classification system.

After properly preparing the input drawings, we used them to train the model. The dataset was divided into three subsets: 80% for training, 10% for validation, and 10% for testing. During the development process, we created multiple datasets and tested various combinations of neural network parameters. The entire process, including the preprocessing of the drawings, was implemented in Python using libraries such as TensorFlow and Keras. The source code was written in Visual Studio Code and continuously saved to a GitHub repository for version control and cloud backup.

4.1.1 Modification of the Input Drawings

Firstly, we resized the width of each input protocol to 2000 pixels. Standardizing the dimensions of all images is important, as it helps the model understand the relative sizes of the groups. Each resized image was then placed on a blank canvas.

In the next step, we centered all images by detecting both auxiliary circles, as shown in Figure 10. Finally, we applied a mask to cover the corner tables, since the program could otherwise mistakenly detect these rectangular tables as sunspot groups.

4.1.2 Detection of Rectangles

Due to variations in scan quality, we applied filters such as blur or contrast enhancement to improve detection. This ensured that any interrupted or faint lines were reconnected. The process is illustrated in Figure 11.

We then used a shape detection algorithm on the modified images. If a shape was identified as a reasonably sized rectangle, it was considered a correctly detected sunspot group. These successfully detected shapes are marked in green in Figure 12.

For each detected group, we extracted a 300×300 pixel region centered on the rectangle and saved it, as shown in Figure 13 (far left). We also stored the coordinates of the rectangle corners in a CSV file for further analysis.

4.1.3 Classification of data

Based on the observation date and time, we calculated the heliographic coordinates of the center and the tilt angle of the Sun's rotational axis for each drawing. Although these values are listed in the observation protocol, it is more efficient to compute them than to extract them using Optical Character Recognition (OCR).

The program also determined the positional angle of each group and its distance from the center of the disk. Using this information and equations (3, 4 and 5), we calculated the heliographic latitude and longitude of each sunspot group.

Subsequently, we identified the corresponding sunspot record by matching heliocentric coordinates with the data table provided by the Astronomical Institute and retrieved its classification according to the McIntosh system.

We then created the required folder structure containing the annotated input data. The main directory included subfolders named after each class, and these subfolders contained the images of the groups corresponding to that classification. From this structure, we could easily select specific groups of interest and use them to train the supervised model.

4.1.4 Data Preparation for Training

To simplify the functioning of the model, we decided to work with black-and-white images. First, the orange-yellow parts of the image (faculae) were programmatically removed. After a manual check, a white mask was applied to each group, covering the surrounding area and leaving only the interior of the rectangle containing the spot. Finally, the image was inverted, as the model performed better with a black background. Figure 13 illustrates the group modification process. Lastly, we divided the prepared data into training, validation, and testing subdirectories.

4.2 Model Architecture

Once we had prepared the dataset structure for the convolutional neural network, we could begin training the individual models. For training all models, we selected the following parameters:

1. Batch size: The number of input samples processed before the model updates.
2. Steps per epoch: The number of training steps in one epoch.
3. Validation steps: The number of steps used for evaluating the model at the end of each epoch.
4. Number and structure of layers: The number and type of layers, the number of neurons in each layer, and their activation functions.

After training, we considered the best model to be the one with the lowest loss function value on the validation data. The final performance was then evaluated on the test data, which the model had not seen during training or validation.

5 RESULTS OF THE WORK

5.1 Classification into Two Classes

5.1.1 Model Axx-Dai

This model is expected to be very accurate in distinguishing between groups Axx and Dai, as the difference between these two groups is enormous. The Axx group consists of a solitary small spot, whereas the Dai group includes penumbrae, is noticeably larger, and, unlike the Axx group, the spots are more distributed towards the poles of the group rather than in its center. Both groups are shown in Figure 14. Additionally, we had hundreds of images of both groups, which also contributes to accuracy. The parameters can be seen in Table 1 and Figure 15:

Number of classes	Total number of samples	Number of samples: Axx, Dai	Batch size	Steps per epoch	Validation steps	Number of layers
2	2499	1668, 831	32	24	38	7

Table 1: Input data and parameters of the CNN model Axx-Dai

Using this combination of parameters, we achieved an overall accuracy of 98.21%, indicating that the model learned to distinguish between the classes with near-perfect precision. This result was expected due to the favorable conditions: clear distinctions between the classes, a large dataset, and a relatively simple classification task.

5.1.2 Model Axx-Bxo

In comparison to the previous model, the Axx-Bxo model is expected to be less accurate because the Axx and Bxo groups differ significantly less from one another. The main distinction between these groups is that Axx represents a single spot, whereas Bxo consists of multiple spots. The appearances of both classes are illustrated in Figure 16. This fact should allow the convolutional neural network to learn to differentiate between them relatively well. Furthermore, we have a substantial amount of data available from both classes, which should also enhance accuracy. The CNN model parameters can be seen in Table 2 and Figure 17:

Number of classes	Total number of samples	Number of samples: Axx, Bxo	Batch size	Steps per epoch	Validation steps	Number of layers
2	3429	1668, 1761	32	20	60	5

Table 2: Input data and parameters of the CNN model Axx-Bxo

Using this combination of parameters, we achieved an overall accuracy of 89.52%. It is evident that the model was able to recognize both categories fairly well. The accuracy could potentially be improved by increasing the number of layers in the network. Another way to enhance performance would be to crop the input images, since both groups are relatively small.

5.2 Classification into Four Classes

Classification is still expected to be highly accurate; however, the models are generally not expected to be more accurate than the two-class models, as accuracy typically decreases with an increasing number of classes.

5.2.1 Model Axx-Csi-Eac-Hsx

In creating the dataset, we intentionally selected four distinct groups, so the neural network should be able to detect the significant differences between them. The Axx group is simply a unipolar spot. The Csi group contains only one spot with a penumbra. The Eac group is the most complex in this dataset, containing both spots with and without penumbrae, with a compact distribution. The Hsx group is also a unipolar spot like Axx, but unlike Axx, it features a penumbra. All class representations are shown in Figure 18. Thus, it is likely that the model will make errors primarily between the Csi and Eac groups or between Axx and Hsx, as these pairs are the least distinct. The parameters of model can be seen in Table 3 and Figure 19:

Number of classes	Total number of samples	Number of samples: Axx, Csi, Eac, Hsx	Batch size	Steps per epoch	Validation steps	Number of layers
4	3991	1668, 572, 206, 1545	20	96	128	7

Table 3: Input data and parameters of the CNN model Axx-Csi-Eac-Hsx

Using this combination of parameters, we achieved an overall accuracy of 92.86%. The convolutional neural network was able to predict the individual classes with high precision, reaching nearly 93% accuracy. Confusion Matrix 1 is shown in Figure 22 (left). Additionally, a clear relationship can be observed between the number of classes and overall accuracy: the Axx-Dai model reached an accuracy of 98.21%. In both cases, we intentionally selected clearly distinct input data.

5.2.2 Model Axx-Bxi-Cai-Cso

The model attempting to classify groups of sunspots into classes Axx, Bxi, Cai, and Cso has a relatively complex task. Unlike the classes of the previous model, these classes do not differ significantly from one another. The groups beginning with the letter C feature spots with a penumbra, which the model should ideally learn to recognize. The distinction between the Cso and Cai groups lies, in addition to the shape of the penumbra, in the arrangement of spots within the group; one group has an open distribution of spots, while the other has a transitional arrangement, meaning spots also appear in the center of the group. The Bxi group contains only umbral spots distributed throughout its area. The Axx group consists of a single unipolar spot. All groups are shown in Figure 20. The parameters can be seen in Table 3 and Figure 21:

Number of classes	Total number of samples	Number of samples: Axx, Bxi, Cai, Cso	Batch size	Steps per epoch	Validation steps	Number of layers
4	4233	1668, 1134, 662, 769	32	14	36	7

Table 4: Input data and parameters of the CNN model Axx-Bxi-Cai-Cso

Using this combination of parameters, we achieved an overall accuracy of 66.05%. It is evident that with this input data, the model did not perform as well as in previous cases. The main issue lies in the data themselves, which are very similar to each other. Looking at Confusion Matrix 2, Figure 22 (right), we can observe that group Axx was predicted correctly 65 times and misclassified only twice. However, class Cai was confused with class Cso in approximately half of the cases. Class Cso was then misclassified evenly as either Bxi or Cso.

5.3 Final Class Model

We also attempted to train a model that included every class. Unfortunately, some classes had very few or even zero samples, making this task impossible. As a result, we excluded 10 out of the 60 possible classes. Due to the drastically uneven number of samples across the remaining classes, this model ultimately achieved extremely low accuracy. If we examine the prediction accuracy of each individual component, we find that the first letter was correctly predicted in 45% of cases, the second letter in only 36%, and the spot distribution (third letter) was predicted with 46% accuracy. Although these numbers may seem somewhat acceptable on their own, combining them results in an overall accuracy of only 8%. This example clearly highlights how critical it is for each component of the model's output to be accurate in order to achieve good overall performance.

5.4 Classification by Letters

Given the challenges we encountered due to the lack of data for specific classes during the model creation, an option arose to train a separate neural network model for each subclassification. This approach dramatically reduces the number of classes that the model predicts and significantly increases the number of samples in each class. We thus created datasets by grouping all samples with a specific subclassification into the same class. For instance, when creating a model that predicts the first letter, all samples from classes Hax, Hhx, Hkx, Hrx, and Hsx would be placed into class H.

5.4.1 Model A-B-C-D-E-F-H

This model should not encounter problems with classification since it has access to a large amount of input data, classifies into a small number of classes, and the differences between the classes are mostly visible, as illustrated in Figure 2. The parameters of CNN model can be seen in Table 5 and Figure 23:

Number of classes	Total number of samples	Number of samples: A, B, C, D, E, F, H	Batch size	Steps per epoch	Validation steps	Number of layers
7	14 768	1668, 2895, 3479, 2872, 1046, 131, 2427	32	28	56	9

Table 5: Input data and parameters of the CNN model A-B-C-D-E-F-H

Using this combination of parameters, we achieved an overall accuracy of 61.22%. The results indicate that the model achieved relatively decent outcomes. Its accuracy is comparable to the Axx-Bxi-Cai-Cso model. By examining Confusion Matrix 1, see Figure 26 (left), we can see that the model's main inaccuracy lies between groups DEF, which we know, only differs in size.

5.4.2 Model a-h-k-r-s-x

It could be expected that the model for the second subclassification will be less accurate than both other letter models since this subclassification refers only to the largest spot, whose relative position within the group is always different, making it difficult to precisely determine where this important spot is located in the image. The parameters for the convolutional network can be seen in Table 6 and Figure 24:

Number of classes	Total number of samples	Number of samples: a, h, k, r, s, x	Batch size	Steps per epoch	Validation steps	Number of layers
6	14 768	3564, 291, 1001, 1280, 3819, 4563	48	64	108	7

Table 6: Input data and parameters of the CNN model a-h-k-r-s-x

Using this combination of parameters, we achieved an overall accuracy of 50.00%. By examining Confusion Matrix 2, see Figure 26 (middle), we can see that the model struggled with two classes. One of these is class h, which was expected due to the lower number of input data. The second class causing problems for the model is class r due to its similarity with class x. Furthermore, these two classes logically do not overlap. The label x can only be used if it pertains to group A or B, while the label r is used in the exact opposite case. This logic could be implemented in the final model, significantly increasing the accuracy.

5.4.3 Model c-i-o-x

We can expect the accuracy of this model to be comparable to the accuracy of predicting the first letter, as there are also noticeable differences between the classes. This subclassification relates to the overall distribution of spots within the group. Therefore, the model should be able to estimate the class based on information such as whether the spots are in the middle, located at the edges, and similar features. The parameters can be seen in Table 7 and Figure 25:

Number of classes	Total number of samples	Number of samples: c, i, o, x	Batch size	Steps per epoch	Validation steps	Number of layers
4	14 768	1603, 4776, 4288, 4101	56	72	108	7

Table 7: Input data and parameters of the CNN model c-i-o-x

Using this combination of parameters, we achieved an overall accuracy of 68.17%. This accuracy is comparable to that of the model for classes A-B-C-D-E-F-H. Confusion Matrix 3, shown in Figure 26 (right), reveals that the model most frequently confused class c with class i, but most of the data points remain close to the main diagonal.

5.5 Final Letter Model

With all three subclassification models ready, we combined them into a single code block. Each model analyzes the same image independently, and their outputs are merged to generate the final group name. The outcomes of these three subclassification models were as follows:

- Correct detection of all three letters: 19.02%
- Correct detection of any two letters: 33.74%
- Correct detection of only one letter: 32.52%

These results show a significant improvement compared to the final class model, as the accuracy more than doubled. Additionally, in Fugure 27, we can observe that some data points tend to cluster around the diagonal in the confusion matrix. However, the model currently lacks the implementation of information regarding which classes do not exist; for example, some test sample was incorrectly labeled as the group Bri.

6 Future Work

6.1 Creation of Additional Models

There are many opportunities to build upon the models developed in this work. Improving sunspot detection in drawings, potentially extracting up to four times more data, could significantly boost accuracy. To access larger datasets, we also explored using drawings from other observatories, such as the Kanzelhöhe Observatory, which kindly shared their data. However, their sunspot groups lack rectangular boundaries and digital labels and use a different classification system, making them incompatible with this project.

An interesting task would be to incorporate information such as the relative position of the group on the disk and its heliographic coordinates into the input data. The model could then better understand the flattening of some groups at the edge of the solar disk. Another potential extension could be to create a program that would work with data labeled with a non-existent class and attempt to correct this information.

Another approach is to convert sunspot group names into CV index values and predict only the group number. This reduces misclassification by keeping similar groups numerically close.

The data could also be used to train a completely different type of model that could predict the evolution of groups over time, as group development over time can be tracked.

6.2 Implementation by Observatories

Once group detection and classification methods are fully functional and flexible, a program could automatically identify and classify sunspot groups from clean drawings. A prototype was developed during a visit of the Ondřejov Observatory, which processed a drawing, detected sunspots, predicted their class, and labeled them. A final version could serve similarly to verify manual classifications.

7 Conclusion

We successfully created a functioning convolutional neural network model capable of classifying groups of sunspots according to the McIntosh system. The work can be understood as proof of concept that machine learning can be applied in a field such as solar astronomy.

The work also includes a section that introduces the problem from the perspective of both branches of science: solar astronomy and machine learning.

Additionally, the work describes the process necessary to create the similar model and explores several other possible directions for future development in this field.