Eclipse Group 02

Matthew Vitullo, Ethan Martinez, Obinna Kalu

Overview of Project:

We were given over 80,000 images of solar eclipses broken up into three separate categories: Total Eclipse, Partial Eclipse, Not an Eclipse. Our group had multiple objectives and tasks. The one main goal however was to build a classifier for these images.

The first being, is there any benefit from using morphological operations on the images. Would we be able to get any features or data that is helpful in determining the kind of eclipse?

The second being developing a PCA and a Manual Classifying Script for the images.

The last one being developing a CNN, in order to classify and predict the images themselves.

Here are our findings and results!

Morphology

Goal: To see if there are any features that we could deduce from using morphological operations that would aid in classifying images. I determined that the way I saw best in classifying these images using features would be using the shape of the moon.

Problem: The largest problem I ran into was glare. The images often had glare in them that would make it hard for the filter to depict the moon itself. Therefore I tried to come up with ways to reduce the glare and give us an idea of a helpful feature.

Erosion

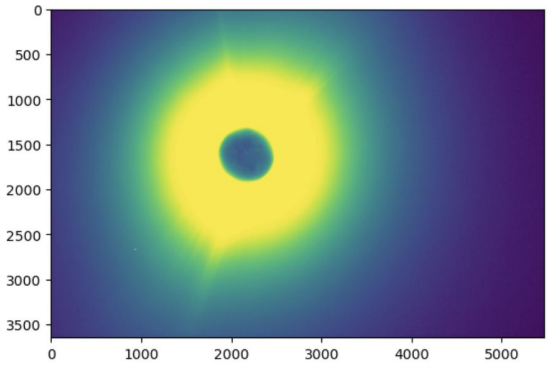
[12]:
```
def apply_erosion(image, n):
    influence_region = np.ones((15,15))

    for i in range(n):
        image = skimage.morphology.erosion(image, footprint=influence_region)


    return image
```

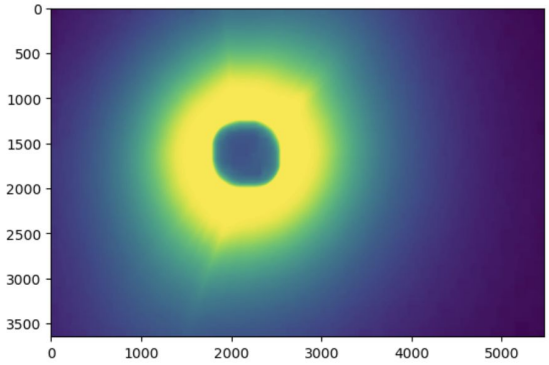
[53]:
```
plt.imshow(apply_erosion(apply_median(images[145]), 0))
```

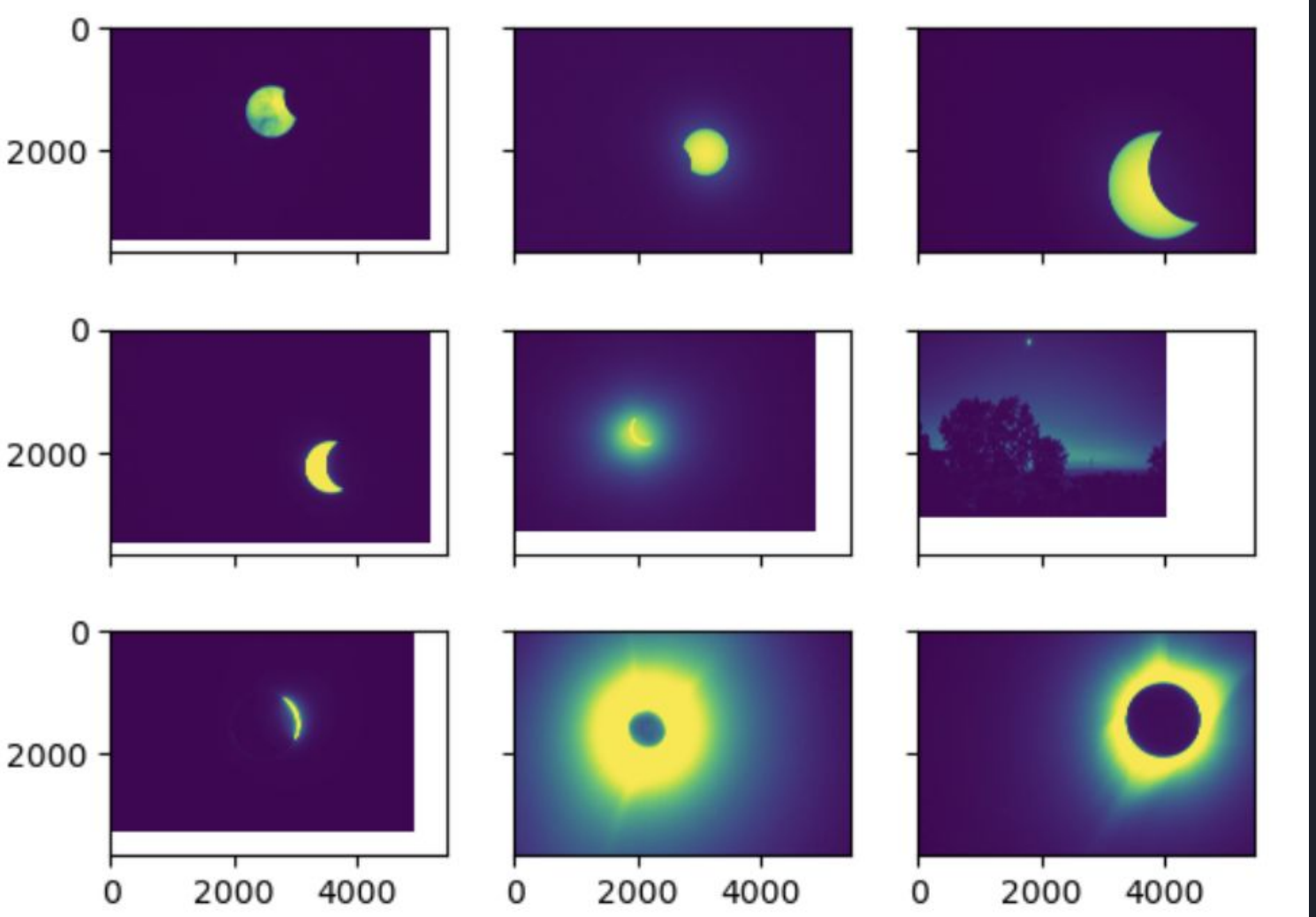
[53]:
```
<matplotlib.image.AxesImage at 0x1309f9450>
```

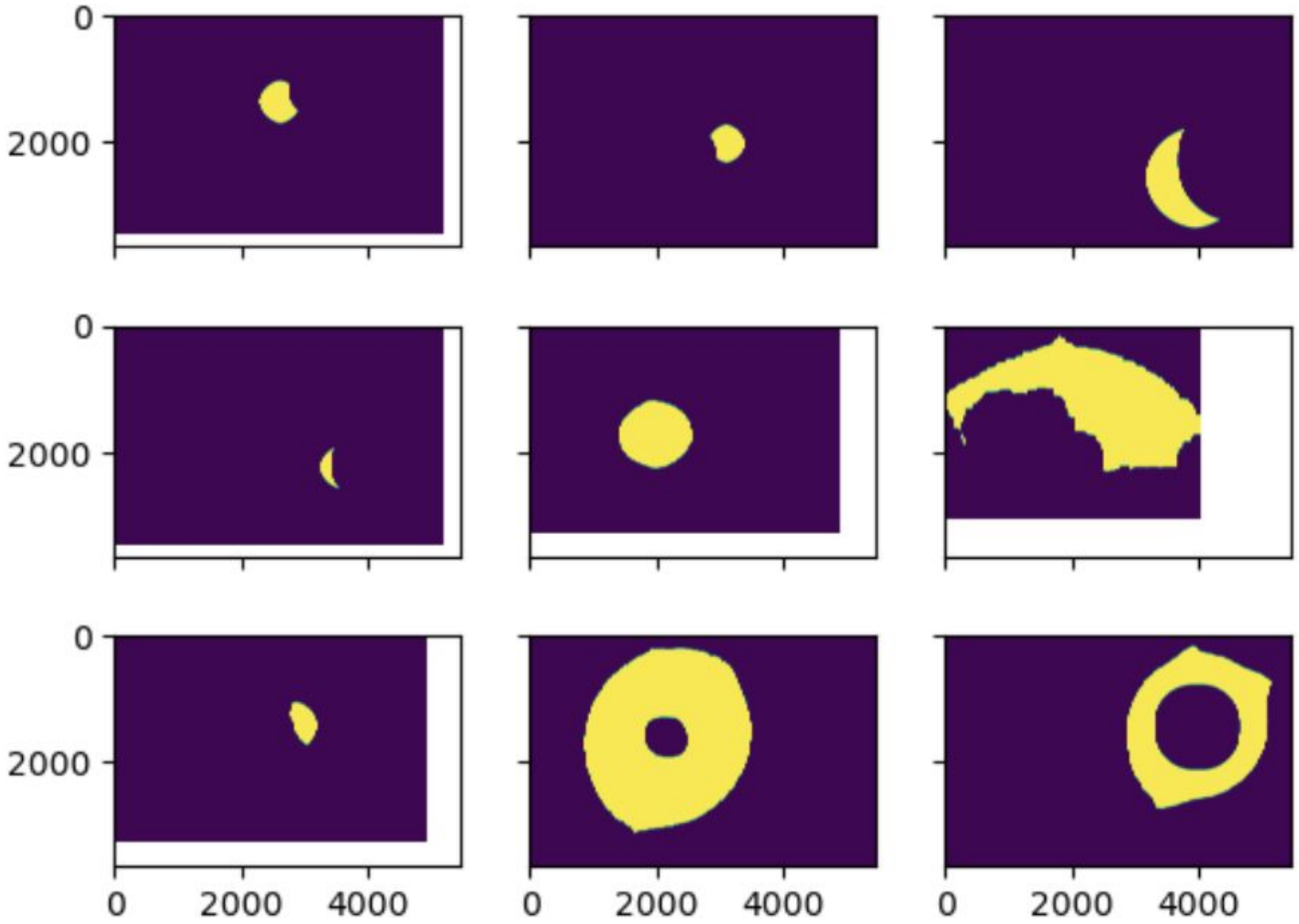


[14]:
```
plt.imshow(apply_erosion(apply_median(images[145]), 10))
```

[14]:
```
<matplotlib.image.AxesImage at 0x14a03af50>
```




0
2000
0
2000
0
2000
0
2000
4000
0
2000
4000
0
2000
4000


0
2000
0
2000
0
2000
0
2000
4000
0
2000
4000
0
2000
4000

Erosion - Results

Purpose: The purpose of erosion is to try and reduce the amount of light space in the image and to increase the amount of dark space. The idea is that we can use the area of the regions in the images to determines.

Result and Conclusion: Erosion although helpful and useful, does not provide a large application of benefit in determining the amount of area for the images. It does enhance the dark space; however, it does not remove the glare.

Histogram of Gradients

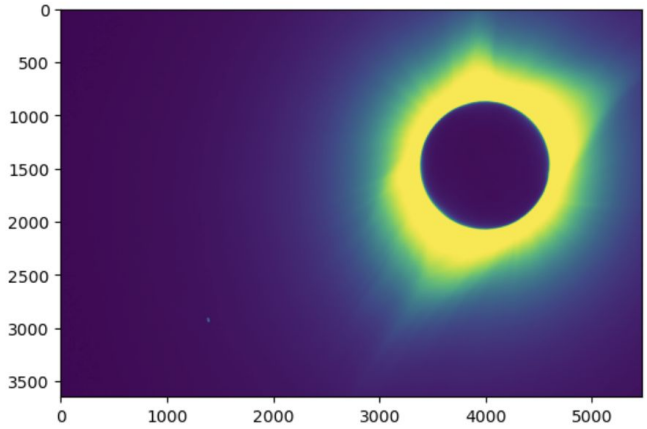
Kaggle describes a Histogram of Gradients as the following: “HOG, or Histogram of Oriented Gradients, is a feature descriptor that is often used to extract features from image data. It is widely used in computer vision tasks for object detection. The technique counts occurrences of gradient orientation in localized portions of an image.”

https://www.kaggle.com/code/brendan45774/hog-features-histogram-of-oriented-gradients

The idea is that we can use this to try and depict the images more accurately.

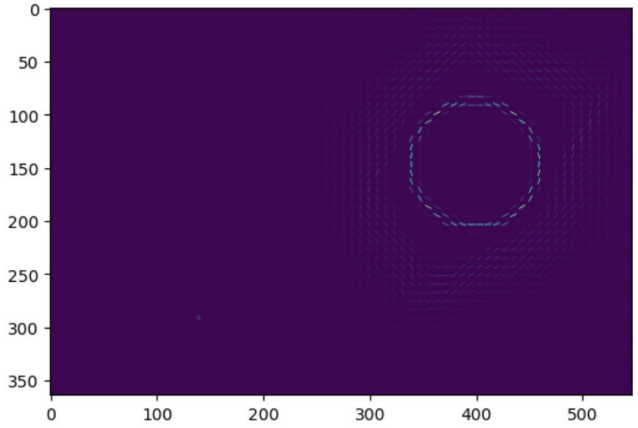
[58]:
```
plt.imshow(apply_median(images[175]))
```

[58]:
```
<matplotlib.image.AxesImage at 0x13233bf40>
```



[57]:
```
plt.imshow(apply_hog(apply_median(images[175])))
```

[57]:
```
<matplotlib.image.AxesImage at 0x1322cc790>
```



Histogram of Gradients - Results

Results: the histogram of gradients is what I find to be the best approach in determining the features or characteristics of the images. It allows us to only get the objects in the image (determined by the edges) and see them as themselves only.

I would say that yes these morphological operations are helpful and are practical in determining the kind of image.

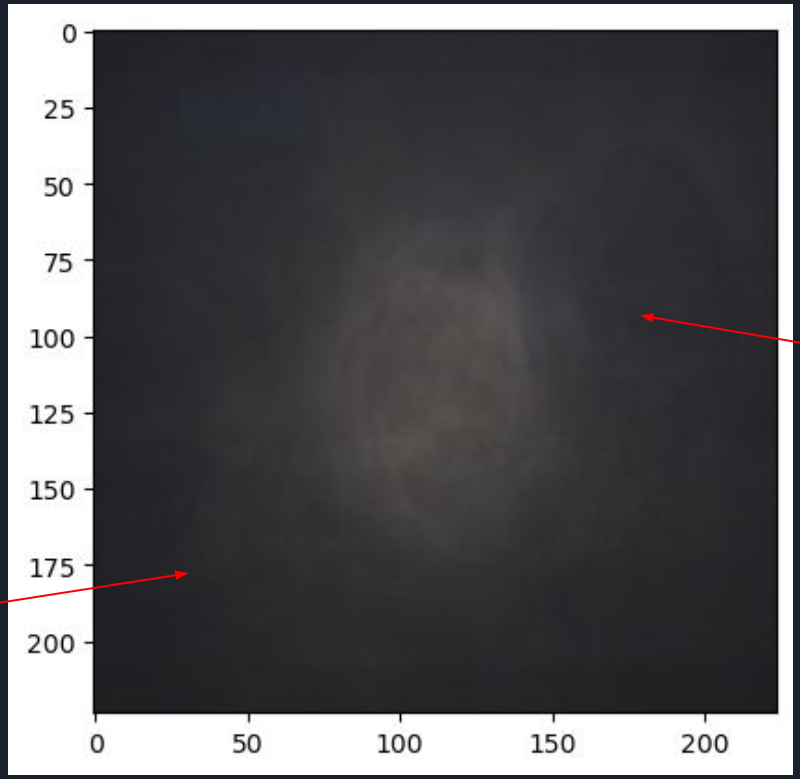
Primary Component Analysis (PCA)

As the website towardsdatascience (https://towardsdatascience.com/feature-extraction-using-principal-component-analysis-a-simplified-visual-demo-e5592ced100a) would explain it, the 'Curse of Dimensionality' is the problem that as a number of features or dimensions grow, the amount of data that needs to be generalized grows exponentially.

PCA works towards mitigating the Curse of Dimensionality, by acting as a dimensionality reduction technique, while preserving as much variance in an image as possible. It works to reduce the number of variables of a data set, while still preserving as much information as possible.

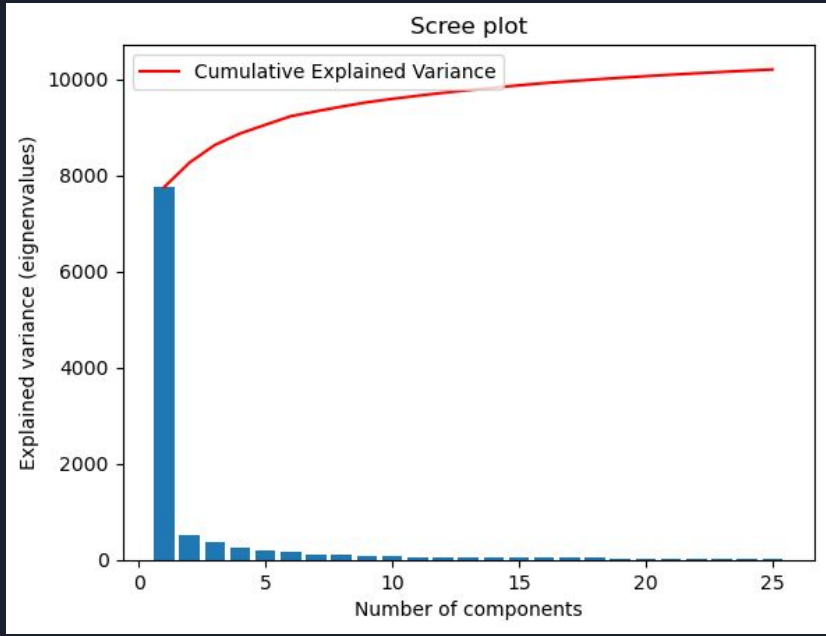
Principal Component Analysis - Results

Average Image generated from the training data:



Principal Component Analysis - Results (Cont)

After generating this Explained Variance chart, it is indicated that a large portion of the variance in the dataset was captured by the first few principal components.



Convolutional Neural Network

- Goal: train a convolutional neural network (CNN) to classify a small subset of approximately 200 images in 8 different classes.
- The scope of the tasks included data preprocessing, model development, training, and evaluation.
- The dataset consisted of the image samples from the 8 classes.
- The results were measured using accuracy, and the performance of the CNN on the test set was evaluated.

Convolutional Neural Network

Approach/Algorithm

- The algorithm implemented was a CNN architecture.
- The architecture consists of several layers including Conv2D, MaxPooling2D, Batch Normalization, Flatten, Dense, and Dropout.
- Parameters such as the number of filters, kernel size, pooling size, activation functions, and dropout rate were set based on trial and error and empirical observations.
- No extensions were made to the original algorithm, but the architecture was customized and fine-tuned to fit the problem domain.)

Convolutional Neural Network

Approach/Algorithm Cont.

Model: "sequential_10"

Layer (type)	Output Shape	Param #
conv2d_31 (Conv2D)	(None, 256, 256, 32)	320
conv2d_32 (Conv2D)	(None, 254, 254, 128)	36992
max_pooling2d_24 (MaxPooling2D)	(None, 127, 127, 128)	0
batch_normalization_24 (BatchNormalization)	(None, 127, 127, 128)	512
conv2d_33 (Conv2D)	(None, 125, 125, 64)	73792
max_pooling2d_25 (MaxPooling2D)	(None, 62, 62, 64)	0
batch_normalization_25 (BatchNormalization)	(None, 62, 62, 64)	256
conv2d_34 (Conv2D)	(None, 60, 60, 128)	73856
max_pooling2d_26 (MaxPooling2D)	(None, 30, 30, 128)	0
batch_normalization_26 (BatchNormalization)	(None, 30, 30, 128)	512
flatten_8 (Flatten)	(None, 115200)	0
dense_16 (Dense)	(None, 128)	14745728
dropout_8 (Dropout)	(None, 128)	0
dense_17 (Dense)	(None, 8)	1032

Total params: 14933000 (56.96 MB)
Trainable params: 14932360 (56.96 MB)
Non-trainable params: 640 (2.50 KB)

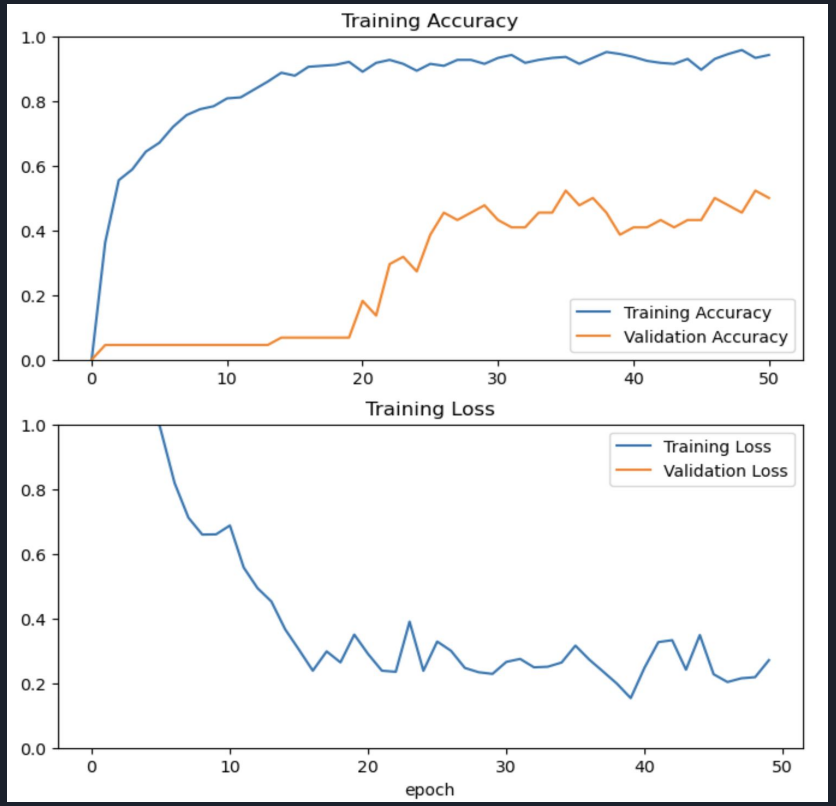
Convolutional Neural Network

Quantitative Results

Accuracy: The model achieved an accuracy of approximately 94% on the training dataset and 50% on the validation dataset.
Loss: The model achieved a loss of 0.27 on the training dataset and 4.67 on the validation dataset.
Learning Rate Reduction: The learning rate was reduced by a factor of 0.5 after 3 consecutive epochs with no improvement in validation loss.

Convolutional Neural Network

Quantitative Results Cont.



Convolutional Neural Network

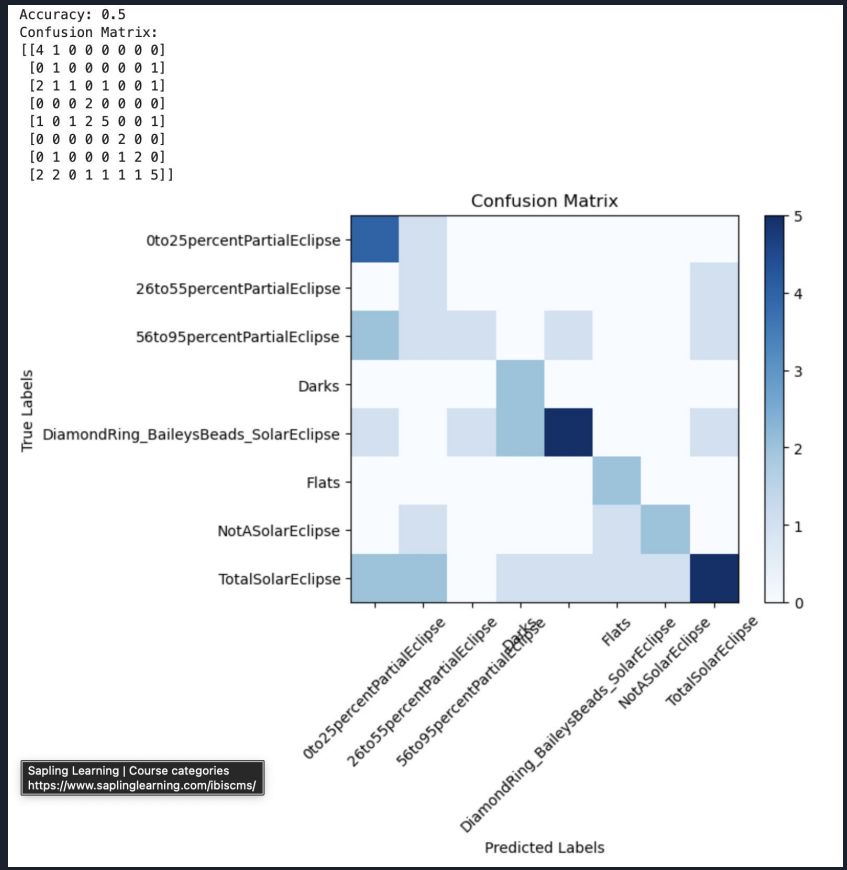
Quantitative Results Cont.

- Overall, the model shows moderate accuracy for some classes but struggles with others
- For example, it performs well on Class 2 and Class 3, with accuracy rates of 48% and 54% respectively. However, it performs poorly on Class 0, Class 1, and Class 5, with accuracy rates of 15%, 26%, and 27% respectively
- These disparities in accuracy are possibly be due to imbalances in the dataset

Convolutional Neural Network

Quantitative Results Cont.

```
Accuracy: 0.5
Confusion Matrix:
[[4 1 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 1]
 [2 1 1 0 1 0 0 1]
 [0 0 0 2 0 0 0 0]
 [1 0 1 2 5 0 0 1]
 [0 0 0 0 0 2 0 0]
 [0 1 0 0 0 1 2 0]
 [2 2 0 1 1 1 1 5]]
```



Convolutional Neural Network

Roadblocks

- HDR/Raw image file types (.cr2, .nef) caused issues with loading these images, further shrinking an already small dataset
- Sunk a lot of time into trying to deal with these raw images (file conversions, error handling, etc.)

Conclusion

Outlook

- CNN Accuracy: The model achieved an accuracy of approximately 94% on the training dataset and 50% on the validation dataset which can be improved with further tuning
- Morphology: The histogram of gradients shows promise and is able to accurately depict the objects in the image as well as isolating outside factors (clouds, glare, etc.).

References

Matthew:

https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_hog.html

https://www.kaggle.com/code/manikg/training-svm-classifier-with-hog-features

https://scikit-image.org/docs/stable/api/skimage.morphology.html#skimage.morphology.erosion

Obinna:

https://datascience.stackexchange.com/questions/93702/cnn-unbalanced-and-small-dataset

https://www.baeldung.com/cs/training-validation-loss-deep-learning

Ethan:

https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

https://www.analyticsvidhya.com/blog/2021/07/svm-and-pca-tutorial-for-beginners/

https://www.visiondummy.com/2014/05/feature-extraction-using-pca/

https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

https://towardsdatascience.com/feature-extraction-using-principal-component-analysis-a-simplified-visual-demo-e5592ced100a