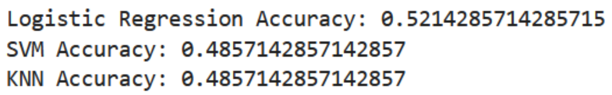
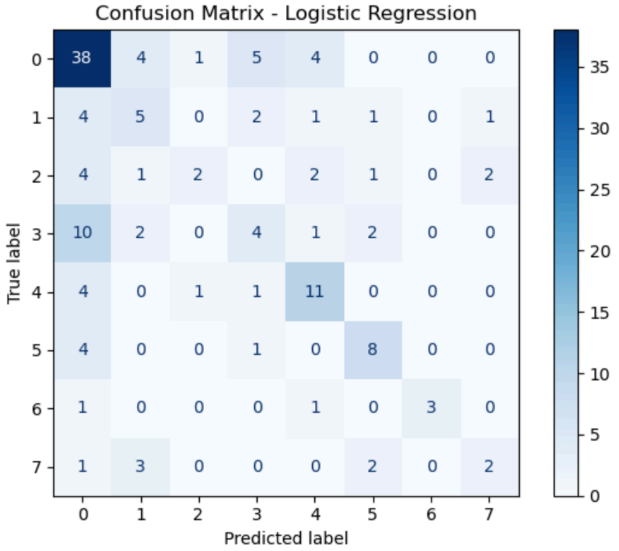
V1 CNN:

First version of our CNN, testing only on logistic regression, SVM, and KNN. No data augmentation was added.

MobileNetV2 results:

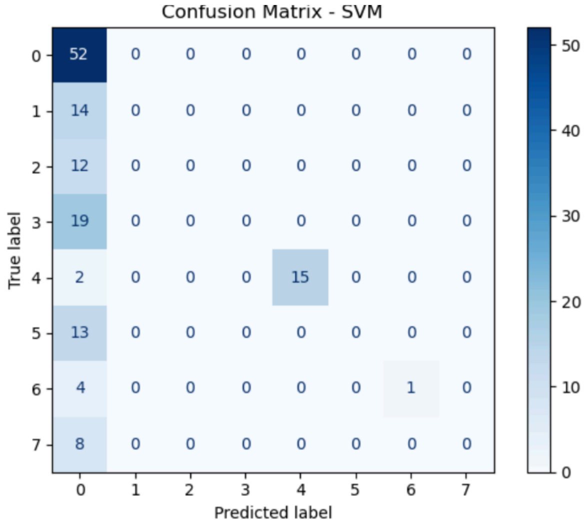


```
Logistic Regression Accuracy: 0.5214285714285715
SVM Accuracy: 0.4857142857142857
KNN Accuracy: 0.4857142857142857
```



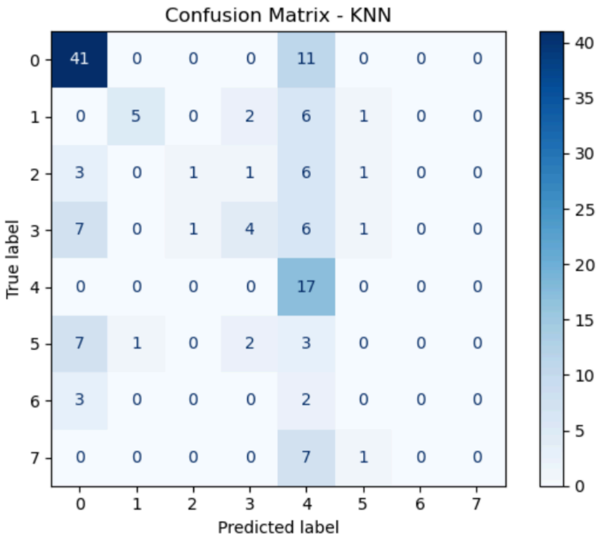
Confusion Matrix - Logistic Regression

True label \ Predicted label	0	1	2	3	4	5	6	7
0	38	4	1	5	4	0	0	0
1	4	5	0	2	1	1	0	1
2	4	1	2	0	2	1	0	2
3	10	2	0	4	1	2	0	0
4	4	0	1	1	11	0	0	0
5	4	0	0	1	0	8	0	0
6	1	0	0	0	1	0	3	0
7	1	3	0	0	0	2	0	2

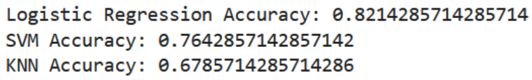


Confusion Matrix - SVM

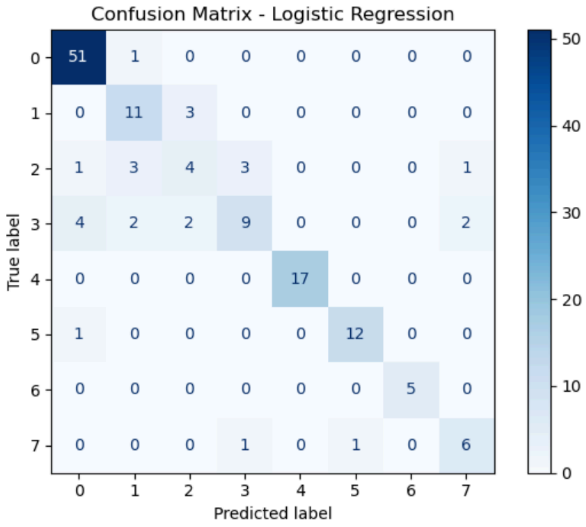
True label \ Predicted label	0	1	2	3	4	5	6	7
0	52	0	0	0	0	0	0	0
1	14	0	0	0	0	0	0	0
2	12	0	0	0	0	0	0	0
3	19	0	0	0	0	0	0	0
4	2	0	0	0	15	0	0	0
5	13	0	0	0	0	0	0	0
6	4	0	0	0	0	0	1	0
7	8	0	0	0	0	0	0	0

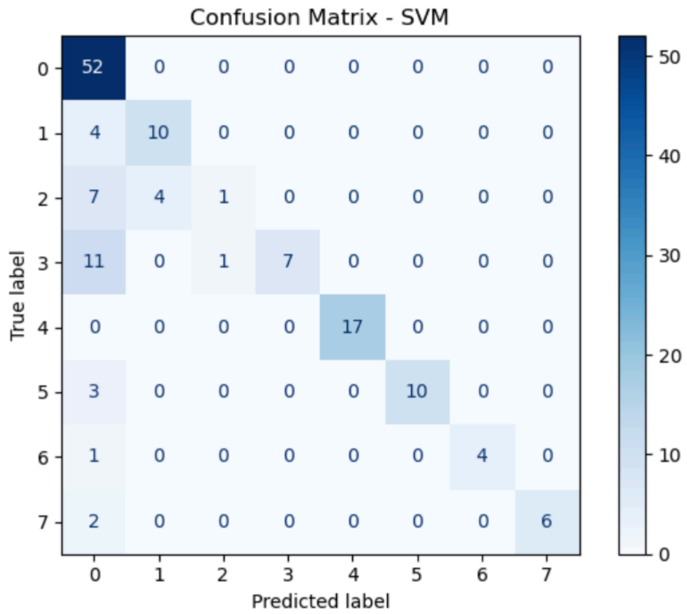


ResNet50 Results:



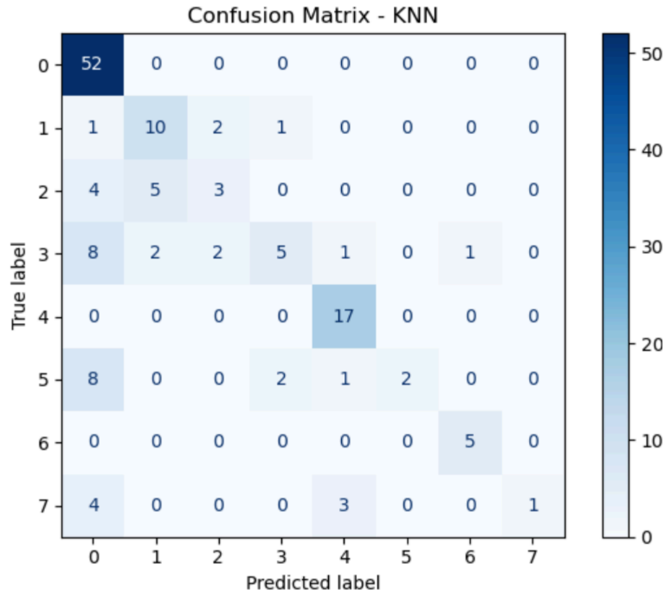
```
Logistic Regression Accuracy: 0.8214285714285714
SVM Accuracy: 0.7642857142857142
KNN Accuracy: 0.6785714285714286
```





Confusion Matrix - SVM

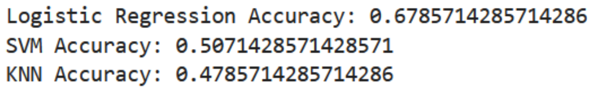
True label \ Predicted label	0	1	2	3	4	5	6	7
0	52	0	0	0	0	0	0	0
1	4	10	0	0	0	0	0	0
2	7	4	1	0	0	0	0	0
3	11	0	1	7	0	0	0	0
4	0	0	0	0	17	0	0	0
5	3	0	0	0	0	10	0	0
6	1	0	0	0	0	0	4	0
7	2	0	0	0	0	0	0	6



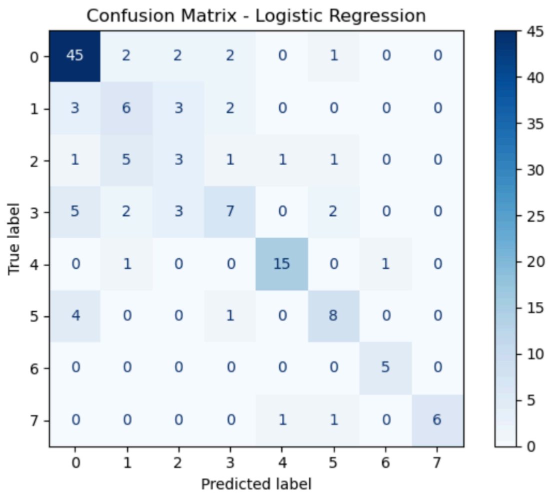
Confusion Matrix - KNN

True label \ Predicted label	0	1	2	3	4	5	6	7
0	52	0	0	0	0	0	0	0
1	1	10	2	1	0	0	0	0
2	4	5	3	0	0	0	0	0
3	8	2	2	5	1	0	1	0
4	0	0	0	0	17	0	0	0
5	8	0	0	2	1	2	0	0
6	0	0	0	0	0	0	5	0
7	4	0	0	0	3	0	0	1

EfficientNetB0 results:

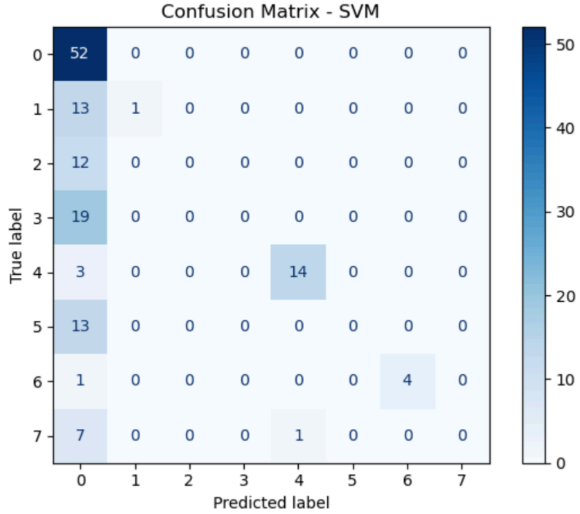


```
Logistic Regression Accuracy: 0.6785714285714286
SVM Accuracy: 0.5071428571428571
KNN Accuracy: 0.4785714285714286
```



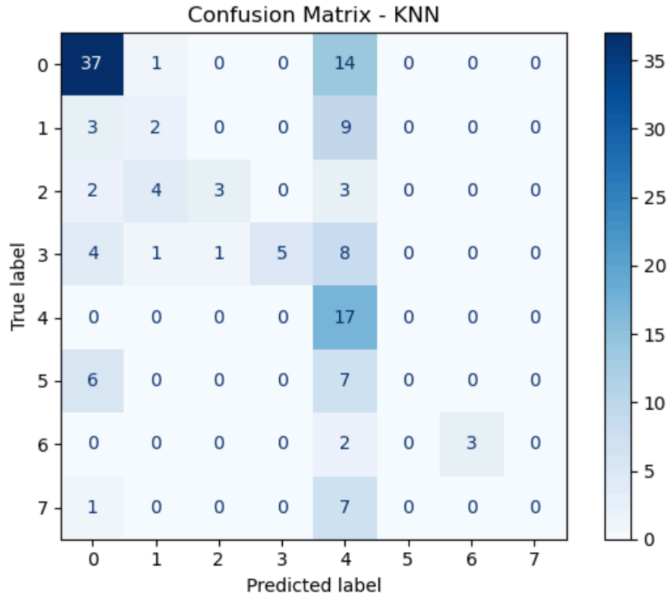
Confusion Matrix - Logistic Regression

	0	1	2	3	4	5	6	7
0	45	2	2	2	0	1	0	0
1	3	6	3	2	0	0	0	0
2	1	5	3	1	1	1	0	0
3	5	2	3	7	0	2	0	0
4	0	1	0	0	15	0	1	0
5	4	0	0	1	0	8	0	0
6	0	0	0	0	0	0	5	0
7	0	0	0	0	1	1	0	6



Confusion Matrix - SVM

	0	1	2	3	4	5	6	7
0	52	0	0	0	0	0	0	0
1	13	1	0	0	0	0	0	0
2	12	0	0	0	0	0	0	0
3	19	0	0	0	0	0	0	0
4	3	0	0	0	14	0	0	0
5	13	0	0	0	0	0	0	0
6	1	0	0	0	0	0	4	0
7	7	0	0	0	1	0	0	0



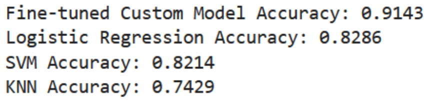
Confusion Matrix - KNN

	0	1	2	3	4	5	6	7
0	37	1	0	0	14	0	0	0
1	3	2	0	0	9	0	0	0
2	2	4	3	0	3	0	0	0
3	4	1	1	5	8	0	0	0
4	0	0	0	0	17	0	0	0
5	6	0	0	0	7	0	0	0
6	0	0	0	0	2	0	3	0
7	1	0	0	0	7	0	0	0

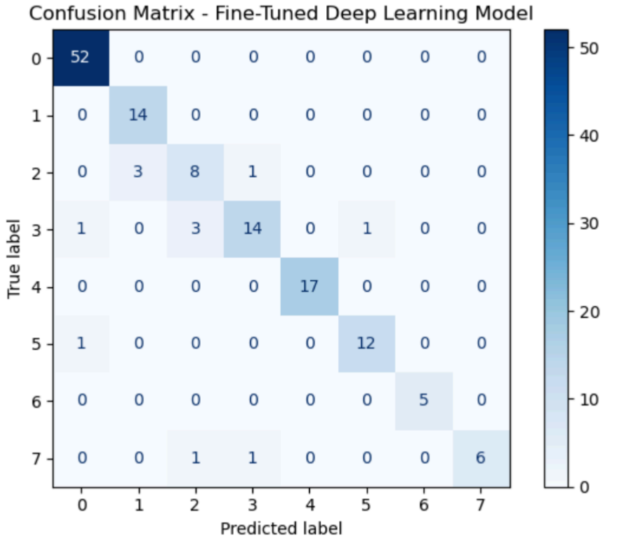
CNN V2 without data augmentation:

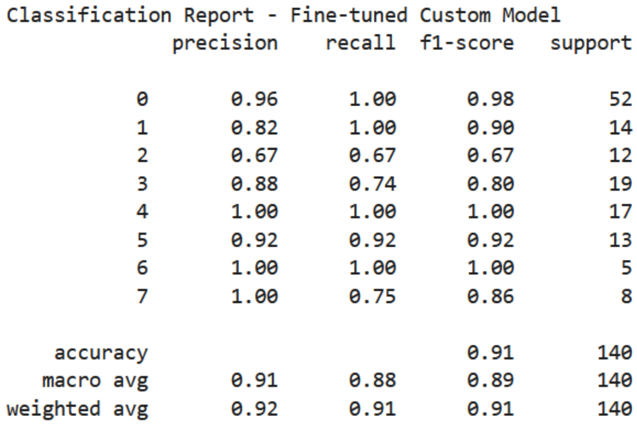
Updated version of CNN now with classification report and misclassified images for fine tuned model.

ResNet50 results:



```
Fine-tuned Custom Model Accuracy: 0.9143
Logistic Regression Accuracy: 0.8286
SVM Accuracy: 0.8214
KNN Accuracy: 0.7429
```

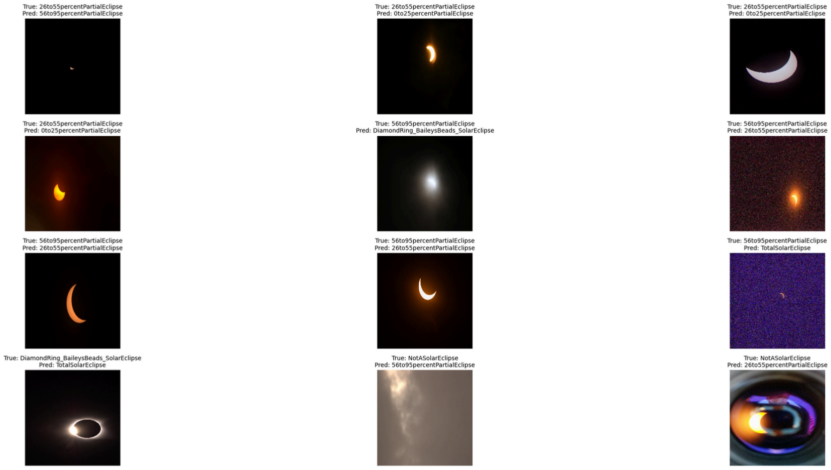




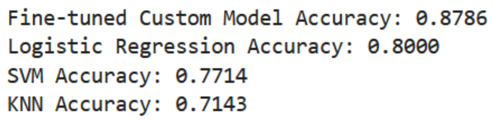
```
Classification Report - Fine-tuned Custom Model
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        52
           1       0.82      1.00      0.90        14
           2       0.67      0.67      0.67        12
           3       0.88      0.74      0.80        19
           4       1.00      1.00      1.00        17
           5       0.92      0.92      0.92        13
           6       1.00      1.00      1.00         5
           7       1.00      0.75      0.86         8

    accuracy                           0.91       140
   macro avg       0.91      0.88      0.89       140
weighted avg       0.92      0.91      0.91       140
```

```
Total misclassified images: 12
```

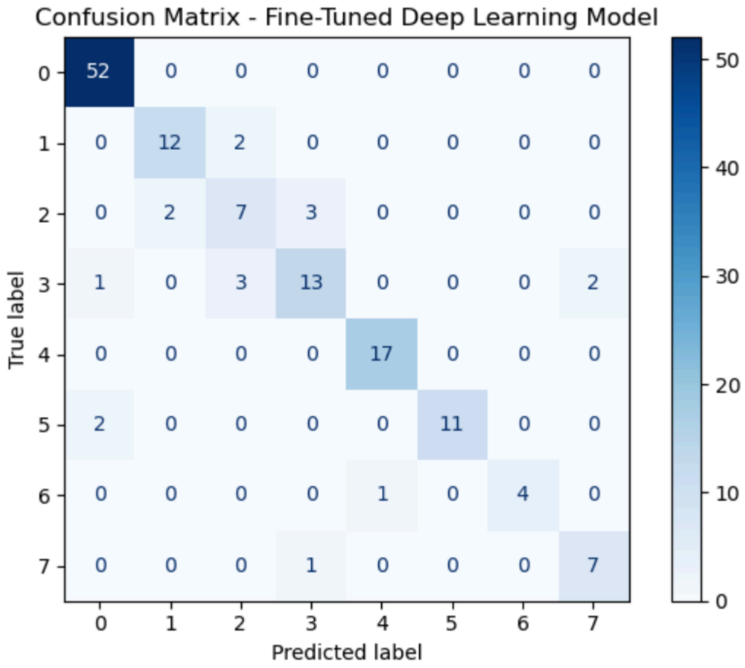
Misclassified Images - Fine-tuned Model


True: 26to55percentPartialEclipse
Pred: 56to95percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 0to25percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 0to25percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 0to25percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: DiamondRing_BaileysBeads_SolarEclipse
True: 56to95percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: TotalSolarEclipse
True: DiamondRing_BaileysBeads_SolarEclipse
Pred: TotalSolarEclipse
True: NotASolarEclipse
Pred: 56to95percentPartialEclipse
True: NotASolarEclipse
Pred: 26to55percentPartialEclipse

EfficientNetB0 results:



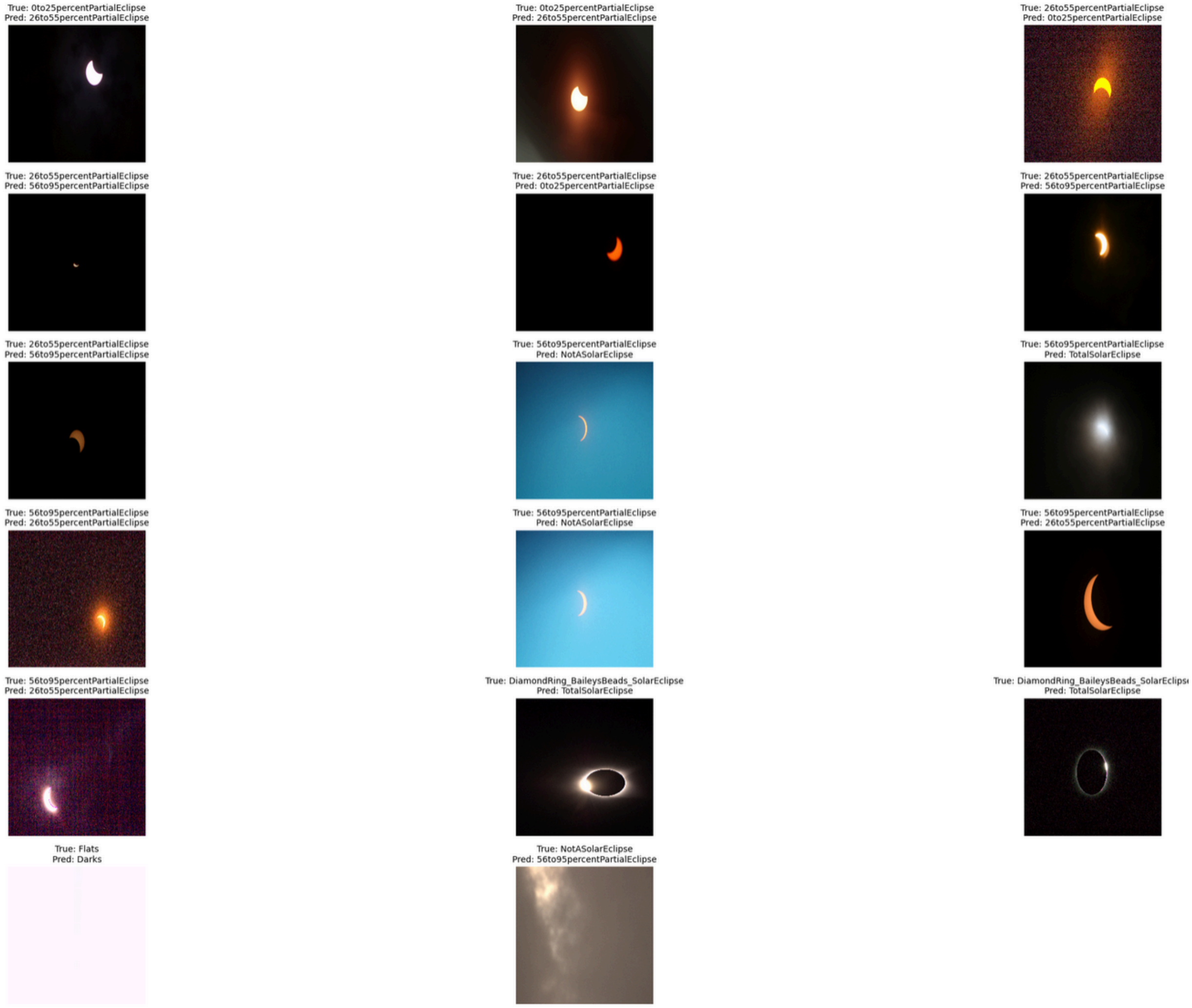
```
Fine-tuned Custom Model Accuracy: 0.8786
Logistic Regression Accuracy: 0.8000
SVM Accuracy: 0.7714
KNN Accuracy: 0.7143
```



	0	1	2	3	4	5	6	7
0	52	0	0	0	0	0	0	0
1	0	12	2	0	0	0	0	0
2	0	2	7	3	0	0	0	0
3	1	0	3	13	0	0	0	2
4	0	0	0	0	17	0	0	0
5	2	0	0	0	0	11	0	0
6	0	0	0	0	1	0	4	0
7	0	0	0	1	0	0	0	7

```
Classification Report - Fine-tuned Custom Model
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        52
           1       0.86      0.86      0.86        14
           2       0.58      0.58      0.58        12
           3       0.76      0.68      0.72        19
           4       0.94      1.00      0.97        17
           5       1.00      0.85      0.92        13
           6       1.00      0.80      0.89         5
           7       0.78      0.88      0.82         8

    accuracy                           0.88       140
   macro avg       0.86      0.83      0.84       140
weighted avg       0.88      0.88      0.88       140
```

```
Total misclassified images: 17
```



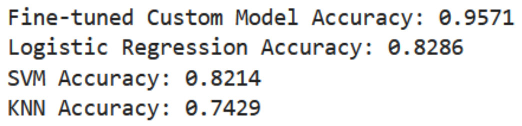
Misclassified Images - Fine-tuned Model

True: 0to25percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 0to25percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 0to25percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 56to95percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 0to25percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 56to95percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 56to95percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: NotASolarEclipse
True: 56to95percentPartialEclipse
Pred: TotalSolarEclipse
True: 56to95percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: NotASolarEclipse
True: 56to95percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: DiamondRing_BaileysBeads_SolarEclipse
Pred: TotalSolarEclipse
True: DiamondRing_BaileysBeads_SolarEclips
Pred: TotalSolarEclipse
True: Flats
Pred: Darks
True: NotASolarEclipse
Pred: 56to95percentPartialEclipse

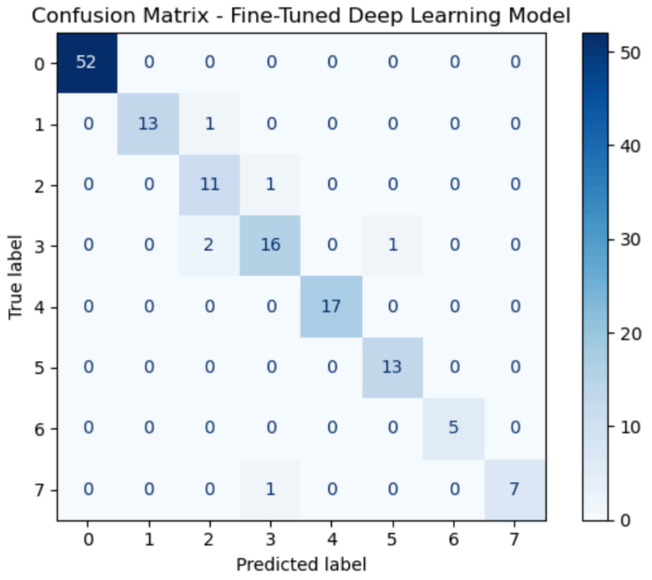
CNN V2 with data augmentation:

Adding data augmentation to the CNN helped improve accuracy a bit and gave us our best results using ResNet50 with our custom model.

ResNet50 Results:

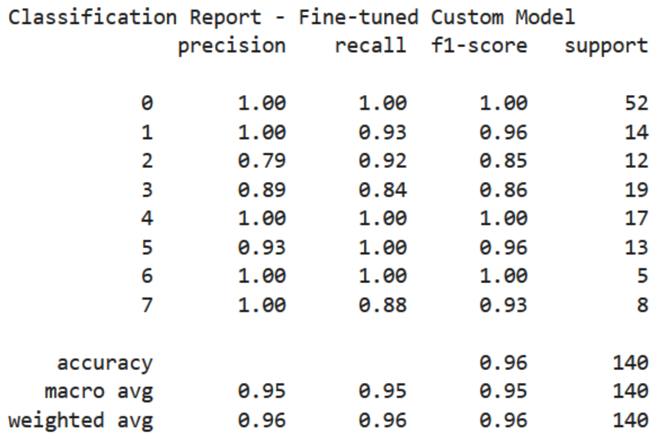


```
Fine-tuned Custom Model Accuracy: 0.9571
Logistic Regression Accuracy: 0.8286
SVM Accuracy: 0.8214
KNN Accuracy: 0.7429
```



Confusion Matrix - Fine-Tuned Deep Learning Model

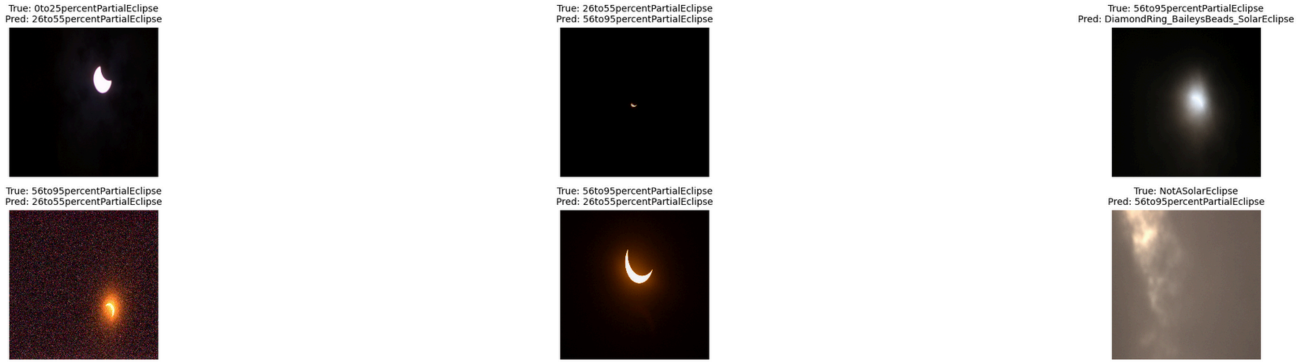
True label \ Predicted label	0	1	2	3	4	5	6	7
0	52	0	0	0	0	0	0	0
1	0	13	1	0	0	0	0	0
2	0	0	11	1	0	0	0	0
3	0	0	2	16	0	1	0	0
4	0	0	0	0	17	0	0	0
5	0	0	0	0	0	13	0	0
6	0	0	0	0	0	0	5	0
7	0	0	0	1	0	0	0	7



```
Classification Report - Fine-tuned Custom Model
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        52
           1       1.00      0.93      0.96        14
           2       0.79      0.92      0.85        12
           3       0.89      0.84      0.86        19
           4       1.00      1.00      1.00        17
           5       0.93      1.00      0.96        13
           6       1.00      1.00      1.00         5
           7       1.00      0.88      0.93         8

    accuracy                           0.96       140
   macro avg       0.95      0.95      0.95       140
weighted avg       0.96      0.96      0.96       140
```

```
Total misclassified images: 6
```

Misclassified Images - Fine-tuned Model


True: 0to25percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 56to95percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: DiamondRing_BaileysBeads_SolarEclipse
True: 56to95percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: NotASolarEclipse
Pred: 56to95percentPartialEclipse

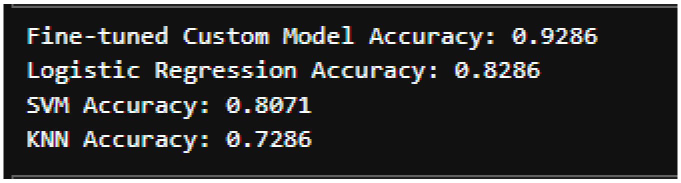
NOTE:

The accuracy varied depending on the machine it was run on.

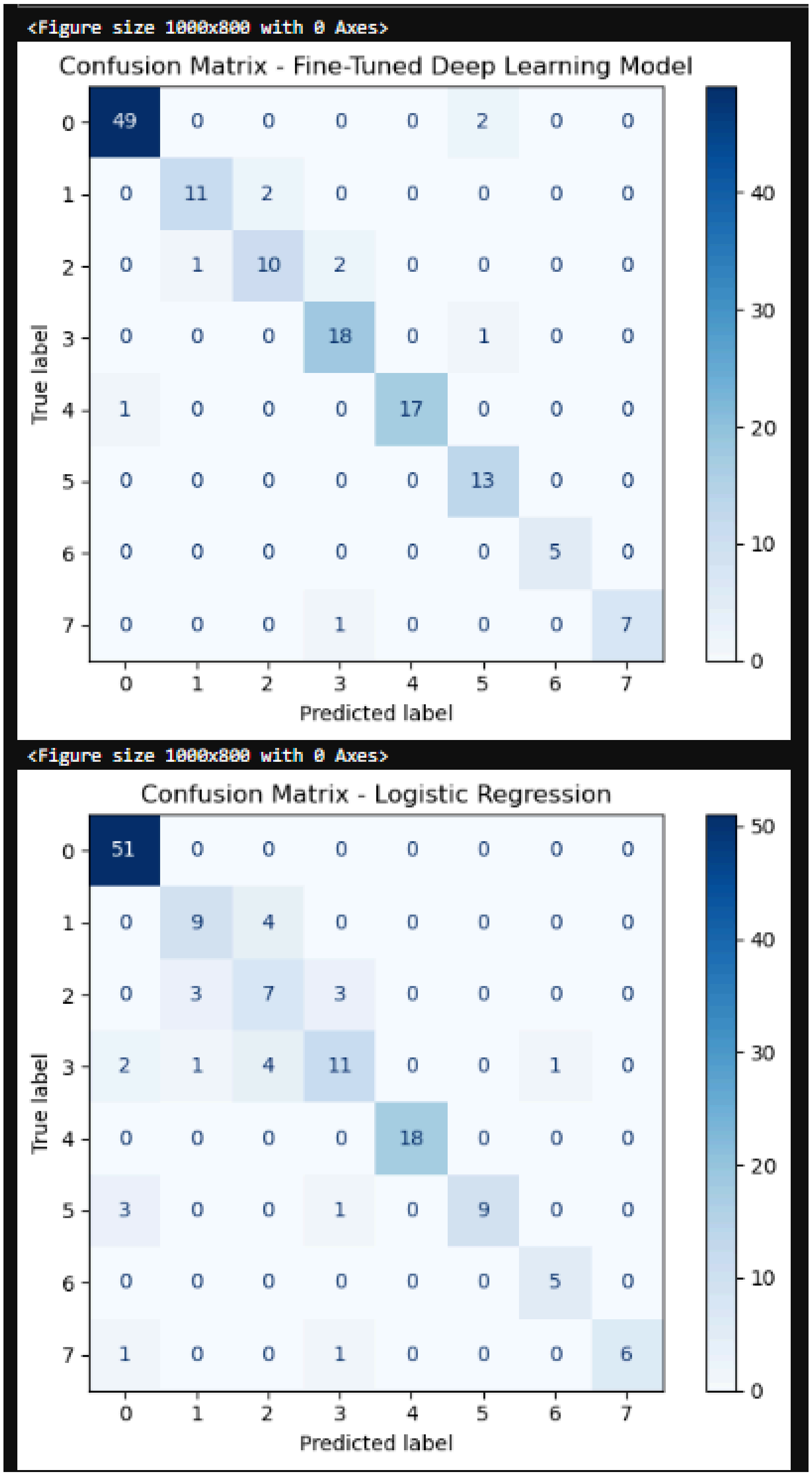
The following shows the accuracies running on a machine with the following specifications:

- Processor: AMD Ryzen 9 5900X 12-Core Processor, 3701 MHz, 12 Core(s), 24 Logical Processors
- RAM: 32 GB
- Graphics Card: EVGA GeForce RTX 3080

Best accuracy: 92.86%



```
Fine-tuned Custom Model Accuracy: 0.9286
Logistic Regression Accuracy: 0.8286
SVM Accuracy: 0.8071
KNN Accuracy: 0.7286
```



```
<Figure size 1000x800 with 0 Axes>
```

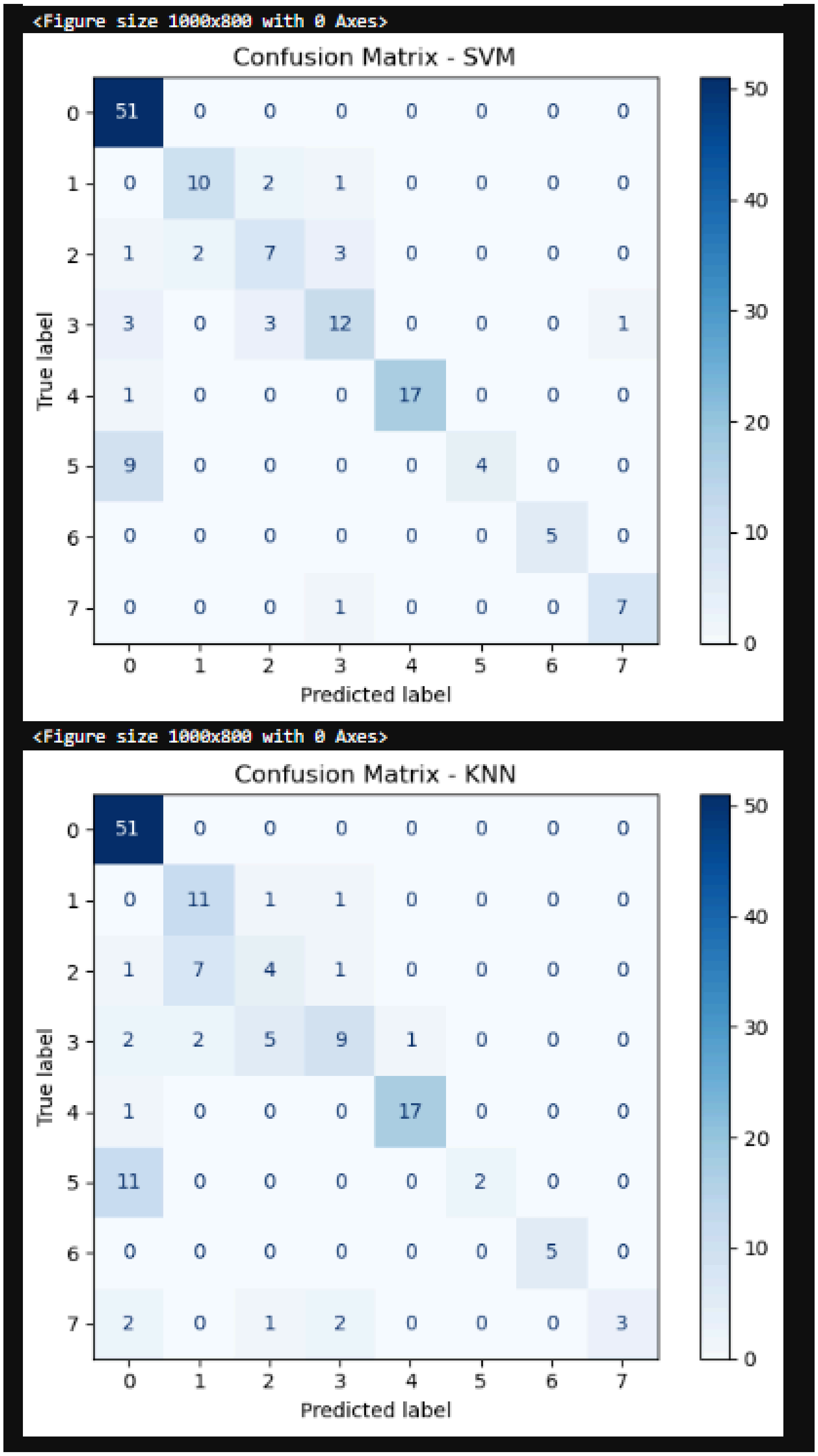
Confusion Matrix - Fine-Tuned Deep Learning Model

	0	1	2	3	4	5	6	7
0	49	0	0	0	0	2	0	0
1	0	11	2	0	0	0	0	0
2	0	1	10	2	0	0	0	0
3	0	0	0	18	0	1	0	0
4	1	0	0	0	17	0	0	0
5	0	0	0	0	0	13	0	0
6	0	0	0	0	0	0	5	0
7	0	0	0	1	0	0	0	7

```
<Figure size 1000x800 with 0 Axes>
```

Confusion Matrix - Logistic Regression

	0	1	2	3	4	5	6	7
0	51	0	0	0	0	0	0	0
1	0	9	4	0	0	0	0	0
2	0	3	7	3	0	0	0	0
3	2	1	4	11	0	0	1	0
4	0	0	0	0	18	0	0	0
5	3	0	0	1	0	9	0	0
6	0	0	0	0	0	0	5	0
7	1	0	0	1	0	0	0	6



```
<Figure size 1000x800 with 0 Axes>
```

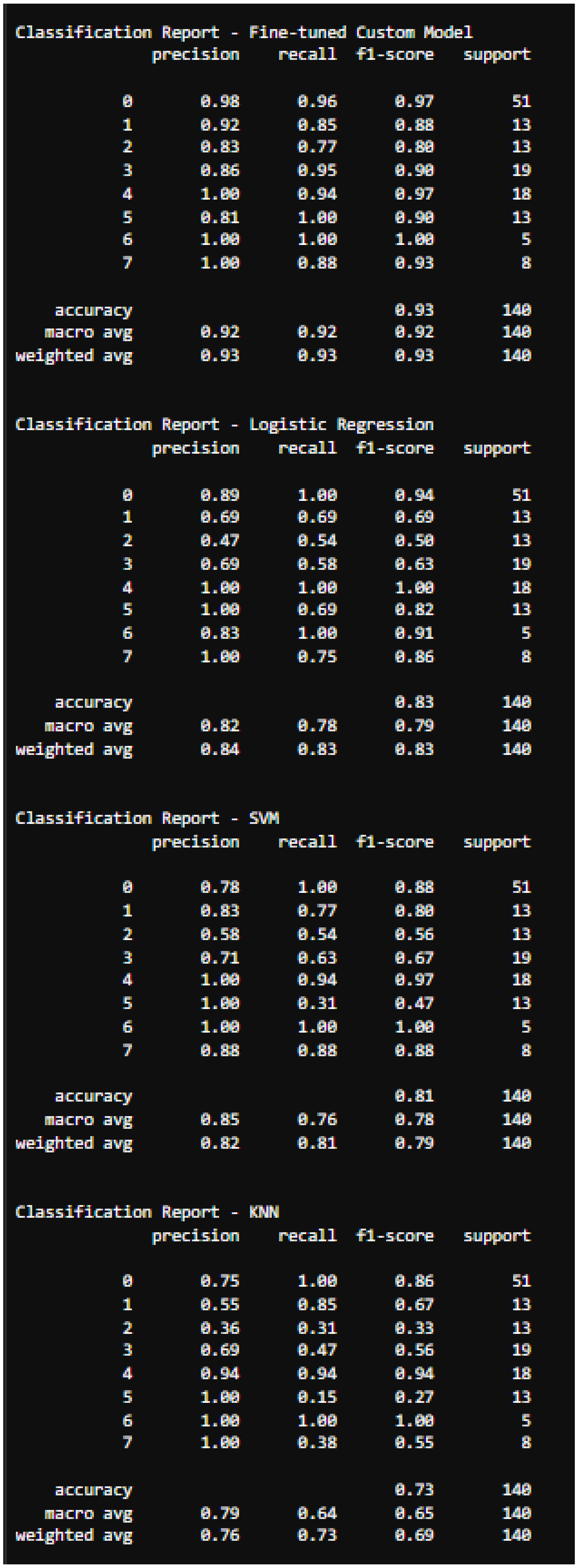
Confusion Matrix - SVM

	0	1	2	3	4	5	6	7
0	51	0	0	0	0	0	0	0
1	0	10	2	1	0	0	0	0
2	1	2	7	3	0	0	0	0
3	3	0	3	12	0	0	0	1
4	1	0	0	0	17	0	0	0
5	9	0	0	0	0	4	0	0
6	0	0	0	0	0	0	5	0
7	0	0	0	1	0	0	0	7

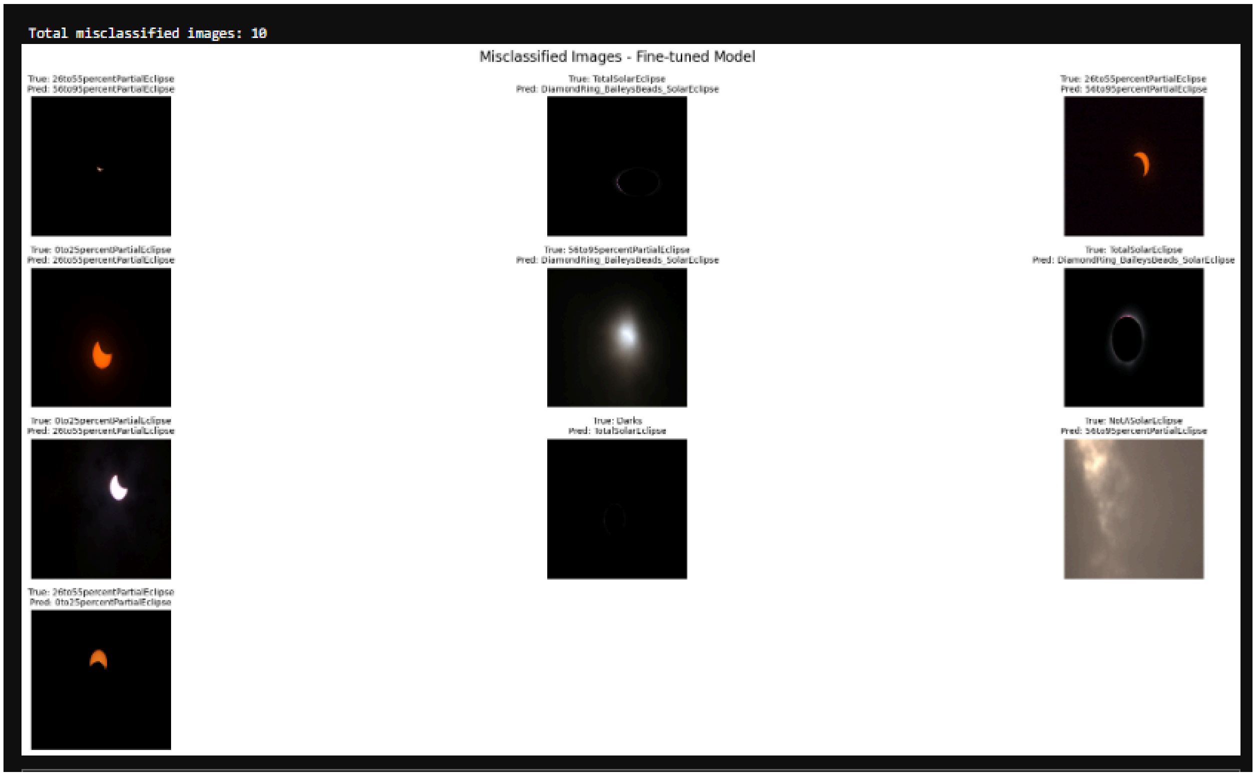
```
<Figure size 1000x800 with 0 Axes>
```

Confusion Matrix - KNN

	0	1	2	3	4	5	6	7
0	51	0	0	0	0	0	0	0
1	0	11	1	1	0	0	0	0
2	1	7	4	1	0	0	0	0
3	2	2	5	9	1	0	0	0
4	1	0	0	0	17	0	0	0
5	11	0	0	0	0	2	0	0
6	0	0	0	0	0	0	5	0
7	2	0	1	2	0	0	0	3



```
Classification Report - Fine-tuned Custom Model
              precision    recall  f1-score   support

           0       0.98      0.96      0.97        51
           1       0.92      0.85      0.88        13
           2       0.83      0.77      0.80        13
           3       0.86      0.95      0.90        19
           4       1.00      0.94      0.97        18
           5       0.81      1.00      0.90        13
           6       1.00      1.00      1.00         5
           7       1.00      0.88      0.93         8

    accuracy                           0.93       140
   macro avg       0.92      0.92      0.92       140
weighted avg       0.93      0.93      0.93       140


Classification Report - Logistic Regression
              precision    recall  f1-score   support

           0       0.89      1.00      0.94        51
           1       0.69      0.69      0.69        13
           2       0.47      0.54      0.50        13
           3       0.69      0.58      0.63        19
           4       1.00      1.00      1.00        18
           5       1.00      0.69      0.82        13
           6       0.83      1.00      0.91         5
           7       1.00      0.75      0.86         8

    accuracy                           0.83       140
   macro avg       0.82      0.78      0.79       140
weighted avg       0.84      0.83      0.83       140


Classification Report - SVM
              precision    recall  f1-score   support

           0       0.78      1.00      0.88        51
           1       0.83      0.77      0.80        13
           2       0.58      0.54      0.56        13
           3       0.71      0.63      0.67        19
           4       1.00      0.94      0.97        18
           5       1.00      0.31      0.47        13
           6       1.00      1.00      1.00         5
           7       0.88      0.88      0.88         8

    accuracy                           0.81       140
   macro avg       0.85      0.76      0.78       140
weighted avg       0.82      0.81      0.79       140


Classification Report - KNN
              precision    recall  f1-score   support

           0       0.75      1.00      0.86        51
           1       0.55      0.85      0.67        13
           2       0.36      0.31      0.33        13
           3       0.69      0.47      0.56        19
           4       0.94      0.94      0.94        18
           5       1.00      0.15      0.27        13
           6       1.00      1.00      1.00         5
           7       1.00      0.38      0.55         8

    accuracy                           0.73       140
   macro avg       0.79      0.64      0.65       140
weighted avg       0.76      0.73      0.69       140
```



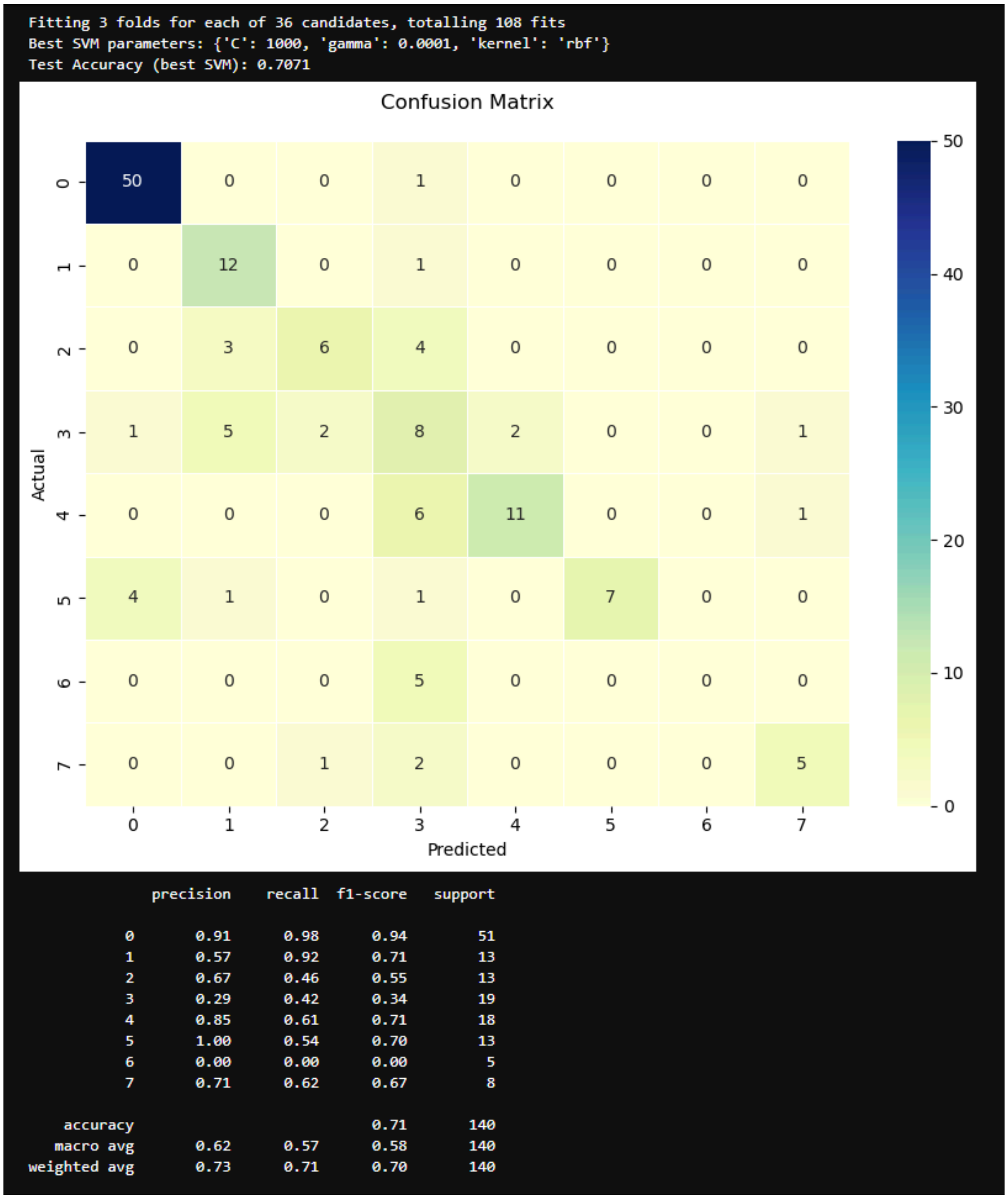
```
Total misclassified images: 10
```

Misclassified Images - Fine-tuned Model
True: 26to55percentPartialEclipse
Pred: 56to95percentPartialEclipse
True: TotalSolarEclipse
Pred: DiamondRing_BaileysBeads_SolarEclipse
True: 26to55percentPartialEclipse
Pred: 56to95percentPartialEclipse
True: 0to25percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: 56to95percentPartialEclipse
Pred: DiamondRing_BaileysBeads_SolarEclipse
True: TotalSolarEclipse
Pred: DiamondRing_BaileysBeads_SolarEclipse
True: 0to25percentPartialEclipse
Pred: 26to55percentPartialEclipse
True: Darks
Pred: TotalSolarEclipse
True: NotASolarEclipse
Pred: 56to95percentPartialEclipse
True: 26to55percentPartialEclipse
Pred: 0to25percentPartialEclipse

SIFT + BoVW + SVM/LR

First Pipeline

Our first attempt at using SIFT and BoVW did not use pre-classification of dark and flat images. Instead we went directly by trying to run all images through SIFT and BoVW and then through SVM and Logistic Regression. The following were our best results:



```
Fitting 3 folds for each of 36 candidates, totalling 108 fits
Best SVM parameters: {'C': 1000, 'gamma': 0.0001, 'kernel': 'rbf'}
Test Accuracy (best SVM): 0.7071
```

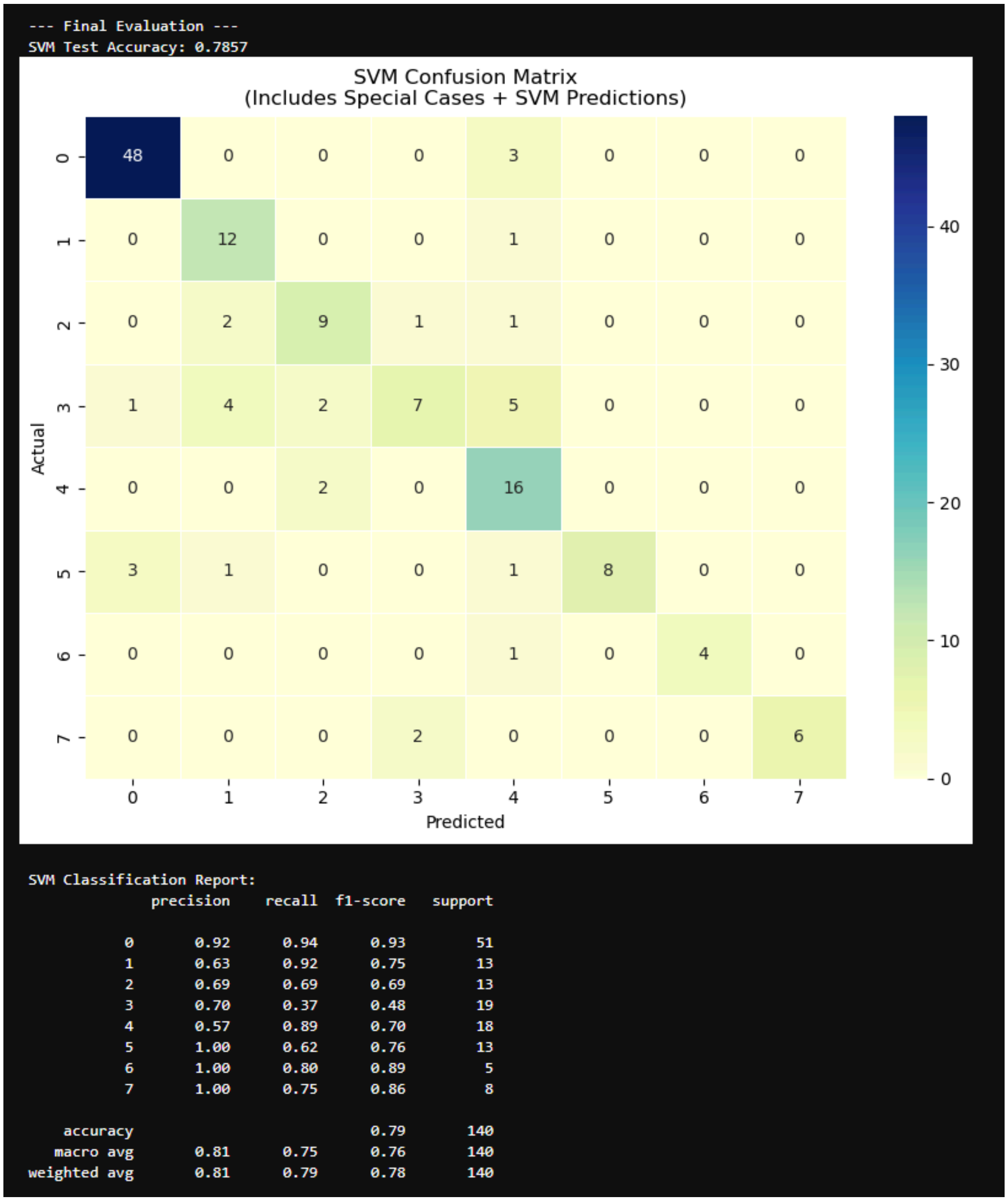
Confusion Matrix

Actual \ Predicted	0	1	2	3	4	5	6	7
0	50	0	0	1	0	0	0	0
1	0	12	0	1	0	0	0	0
2	0	3	6	4	0	0	0	0
3	1	5	2	8	2	0	0	1
4	0	0	0	6	11	0	0	1
5	4	1	0	1	0	7	0	0
6	0	0	0	5	0	0	0	0
7	0	0	1	2	0	0	0	5

```
               precision    recall  f1-score   support

           0       0.91      0.98      0.94        51
           1       0.57      0.92      0.71        13
           2       0.67      0.46      0.55        13
           3       0.29      0.42      0.34        19
           4       0.85      0.61      0.71        18
           5       1.00      0.54      0.70        13
           6       0.00      0.00      0.00         5
           7       0.71      0.62      0.67         8

    accuracy                           0.71       140
   macro avg       0.62      0.57      0.58       140
weighted avg       0.73      0.71      0.70       140
```

Second Pipeline

No matter what parameters we changed we could not get a higher accuracy. So we decided to look at what images were mostly being misclassified based on the confusion matrix and we can see that 6(flats) had no correct predictions and 4(darks) could do better. We decided that we could classify these two beforehand since the key feature in each of these classes was simply the unbalanced histograms. Darks would have more of their peaks towards 0 and flats (brighter images) would have more of their peaks closer to 255. Using this information and intensities of these images, along with the dominant pixel value, we went ahead and tried classifying these images before running the rest of the images through SIFT, BoVW, and SVM/LR. With this we got the following results (best):

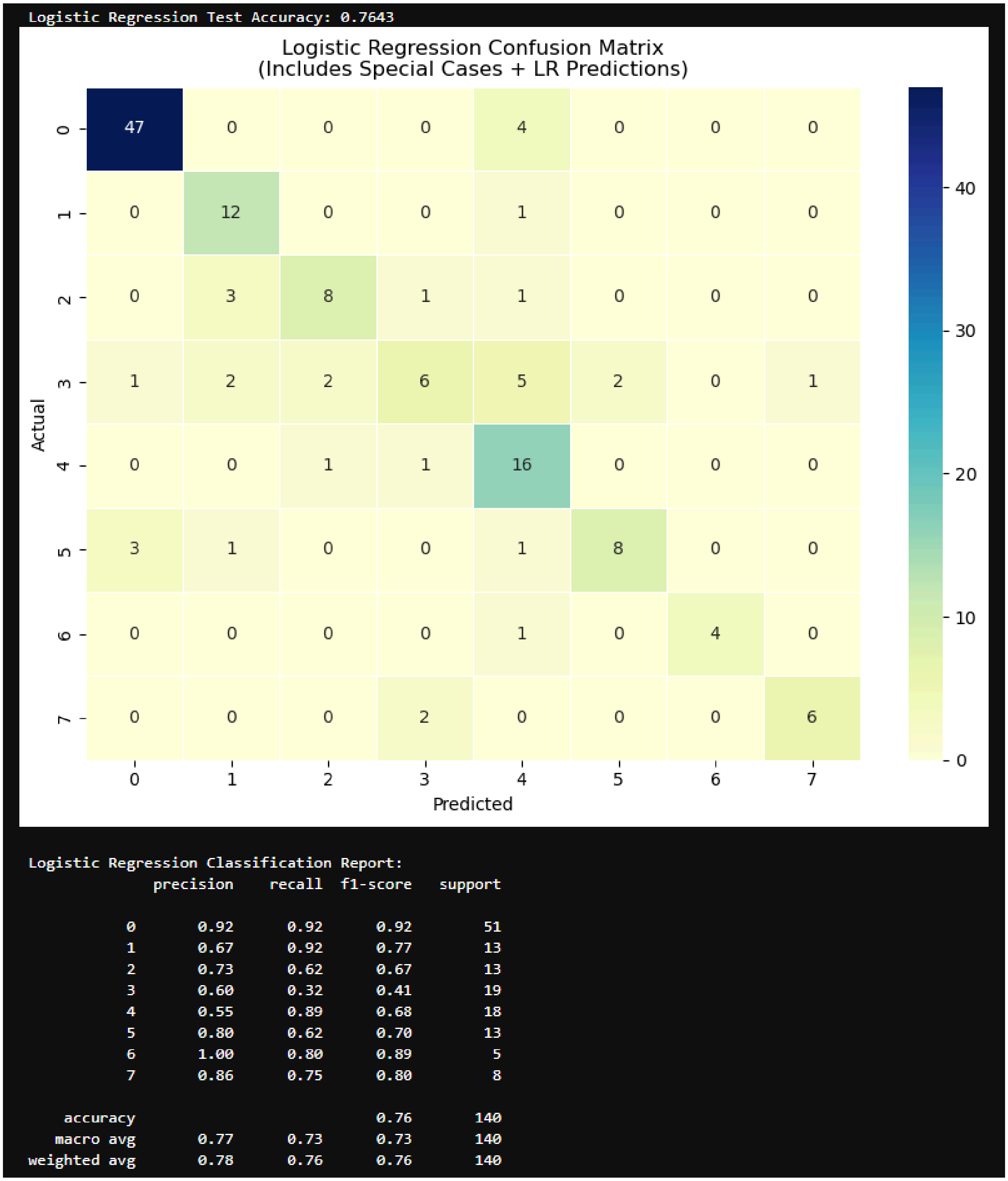


```
--- Final Evaluation ---
SVM Test Accuracy: 0.7857
```

SVM Confusion Matrix
(Includes Special Cases + SVM Predictions)

Actual \ Predicted	0	1	2	3	4	5	6	7
0	48	0	0	0	3	0	0	0
1	0	12	0	0	1	0	0	0
2	0	2	9	1	1	0	0	0
3	1	4	2	7	5	0	0	0
4	0	0	2	0	16	0	0	0
5	3	1	0	0	1	8	0	0
6	0	0	0	0	1	0	4	0
7	0	0	0	2	0	0	0	6

```
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.94      0.93        51
           1       0.63      0.92      0.75        13
           2       0.69      0.69      0.69        13
           3       0.70      0.37      0.48        19
           4       0.57      0.89      0.70        18
           5       1.00      0.62      0.76        13
           6       1.00      0.80      0.89         5
           7       1.00      0.75      0.86         8

    accuracy                           0.79       140
   macro avg       0.81      0.75      0.76       140
weighted avg       0.81      0.79      0.78       140
```



```
Logistic Regression Test Accuracy: 0.7643
```

Logistic Regression Confusion Matrix
(Includes Special Cases + LR Predictions)

Actual \ Predicted	0	1	2	3	4	5	6	7
0	47	0	0	0	4	0	0	0
1	0	12	0	0	1	0	0	0
2	0	3	8	1	1	0	0	0
3	1	2	2	6	5	2	0	1
4	0	0	1	1	16	0	0	0
5	3	1	0	0	1	8	0	0
6	0	0	0	0	1	0	4	0
7	0	0	0	2	0	0	0	6

```
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.92      0.92        51
           1       0.67      0.92      0.77        13
           2       0.73      0.62      0.67        13
           3       0.60      0.32      0.41        19
           4       0.55      0.89      0.68        18
           5       0.80      0.62      0.70        13
           6       1.00      0.80      0.89         5
           7       0.86      0.75      0.80         8

    accuracy                           0.76       140
   macro avg       0.77      0.73      0.73       140
weighted avg       0.78      0.76      0.76       140
```

Overview of what was tried (for SIFT/BoVW/SVM/LR) and what the outcomes were:

- As stated, first we did **not try dark and flat classifying before SIFT**. Everything was running through sift and the **best accuracy** we were able to get (with varying ALL parameters) was **about 70%**.
- **Now pre-classifying flats and darks** through statistical thresholding, bumped the accuracy to our **best accuracy** of **about 79%**.
- With **this approach** our accuracy **varied from 74% up to 79%**. We tried **sharpening** the images since SIFT looks for keypoints that come from corners and edges (for example) because some eclipse images had good edges and points that stuck out, but we actually got a **worse accuracy** even with pre-classification i was getting **around 62%**.
- We tried using a **different preprocessing method for dark/flat classification** than for sift but the accuracy went down to **66%**. I believe we had tried **no CLAHE for flat/dark classification** and **no Gaussian Blur for SIFT/BoVW**.
- Parameters we played with and added:
    - sift = cv2.SIFT_create(**contrastThreshold=0.03, edgeThreshold=5**)
    - **Image size**
    - **Threshold values**
        - def classify_flat_or_dark(img, **flat_std_thresh=8, dark_mean_thresh=25, dark_std_thresh=6**, path=""):
    - **Minikmeans** instead of regular KMeans
        - Lowered the accuracy
    - **Vocab size**
        - Best: 345
        - Larger: Lowered the accuracy
        - Smaller: Lowered the accuracy
    - Added **grid search**
        - **Improved by about 1%** as our parameters before were almost all the same except for gamma, but always good to have