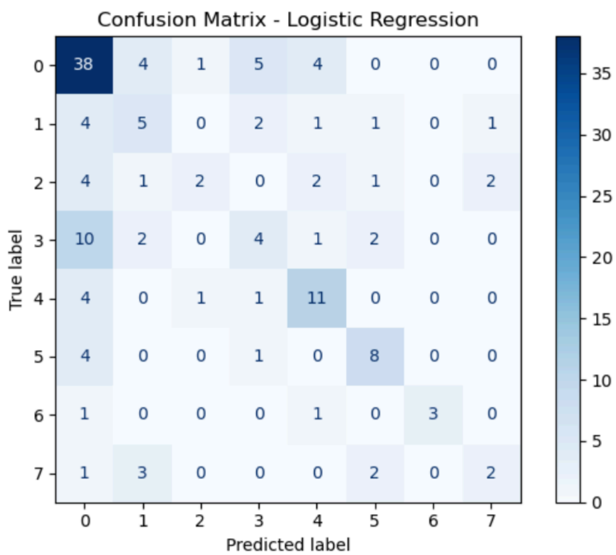
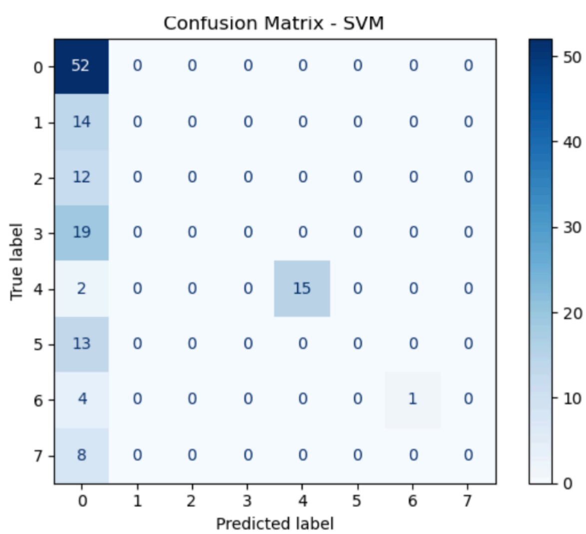
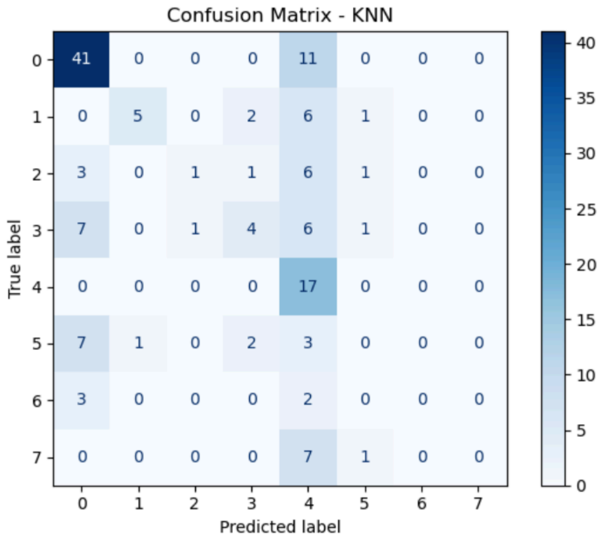
V1 CNN:
MobileNetV2 results:

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

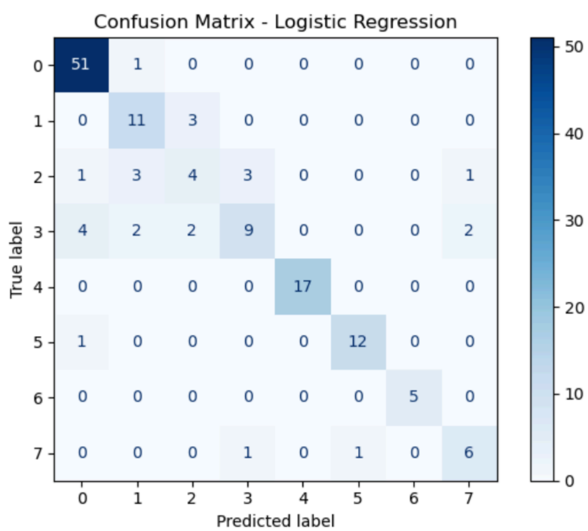


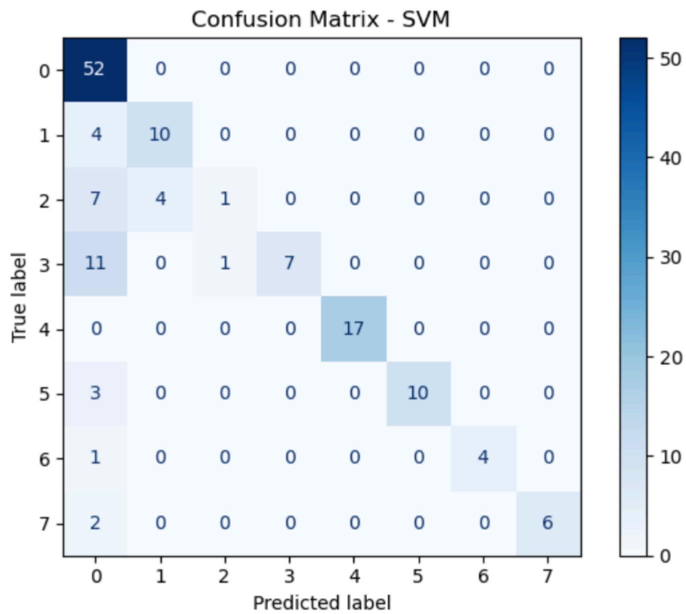


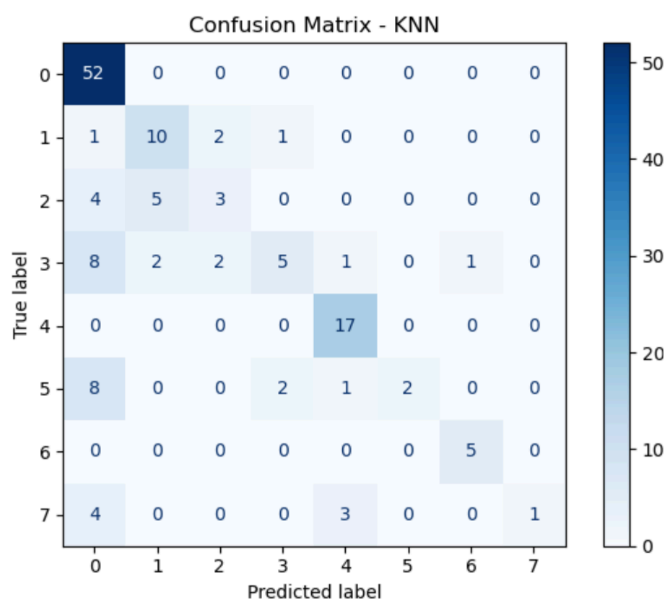


ResNet50 Results:

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

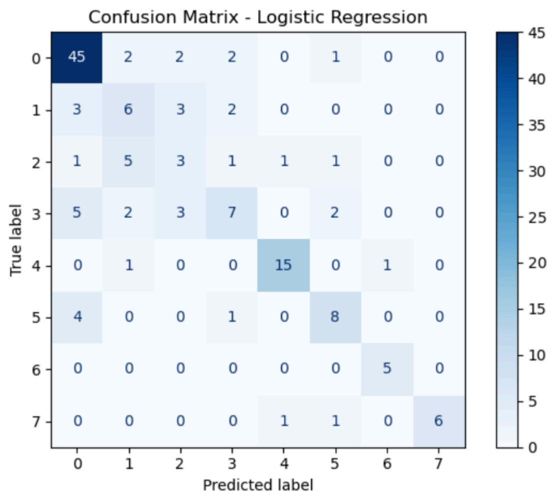


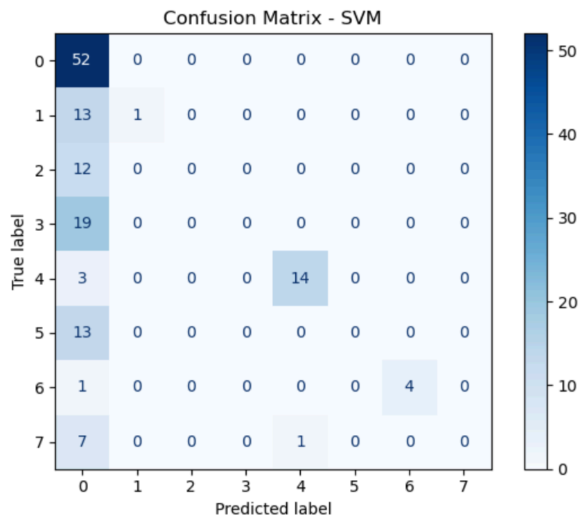


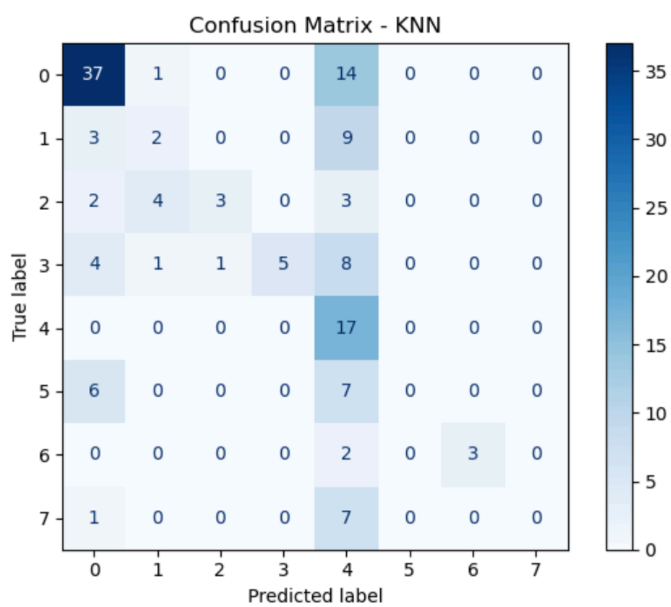


EfficientNetB0 results:

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



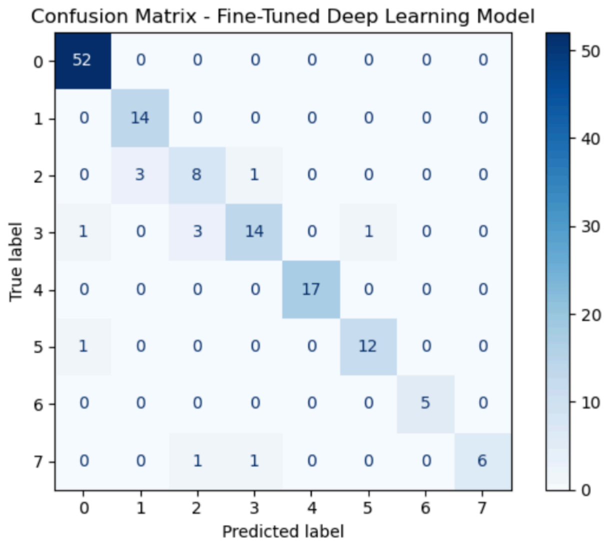




CNN V2 without data augmentation:

ResNet50 results:

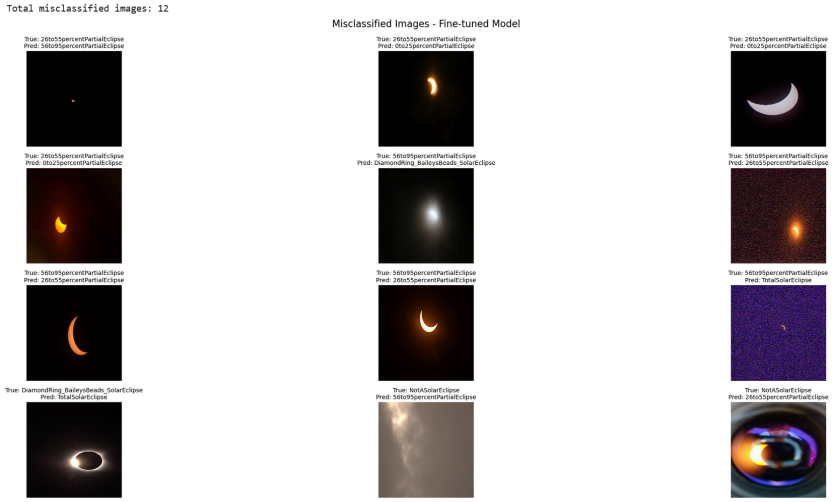
```
Fine-tuned Custom Model Accuracy: 0.9143
Logistic Regression Accuracy: 0.8286
SVM Accuracy: 0.8214
KNN Accuracy: 0.7429
```



Classification Report - Fine-tuned Custom Model

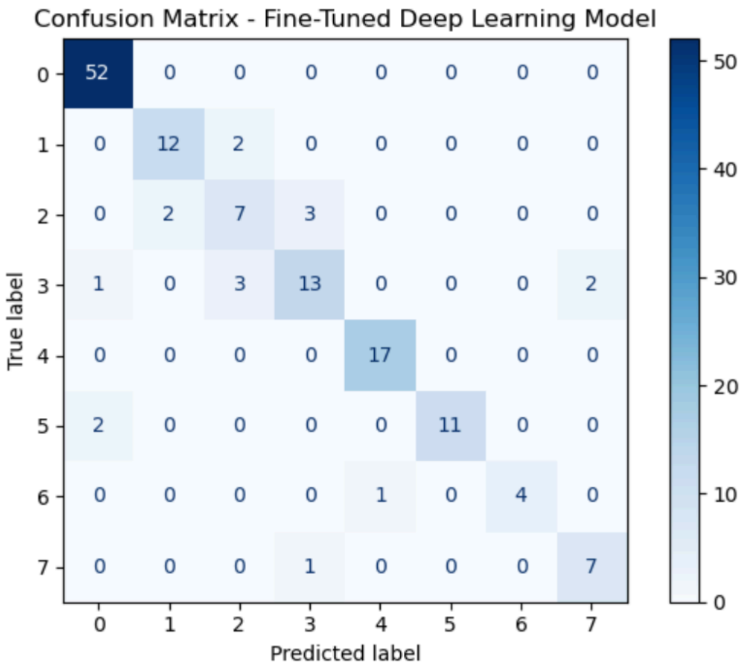
	precision	recall	f1-score	support
0	0.96	1.00	0.98	52
1	0.82	1.00	0.90	14
2	0.67	0.67	0.67	12
3	0.88	0.74	0.80	19
4	1.00	1.00	1.00	17
5	0.92	0.92	0.92	13
6	1.00	1.00	1.00	5
7	1.00	0.75	0.86	8
accuracy			0.91	140
macro avg	0.91	0.88	0.89	140
weighted avg	0.92	0.91	0.91	140

```
Total misclassified images: 12
```



EfficientNetB0 results:

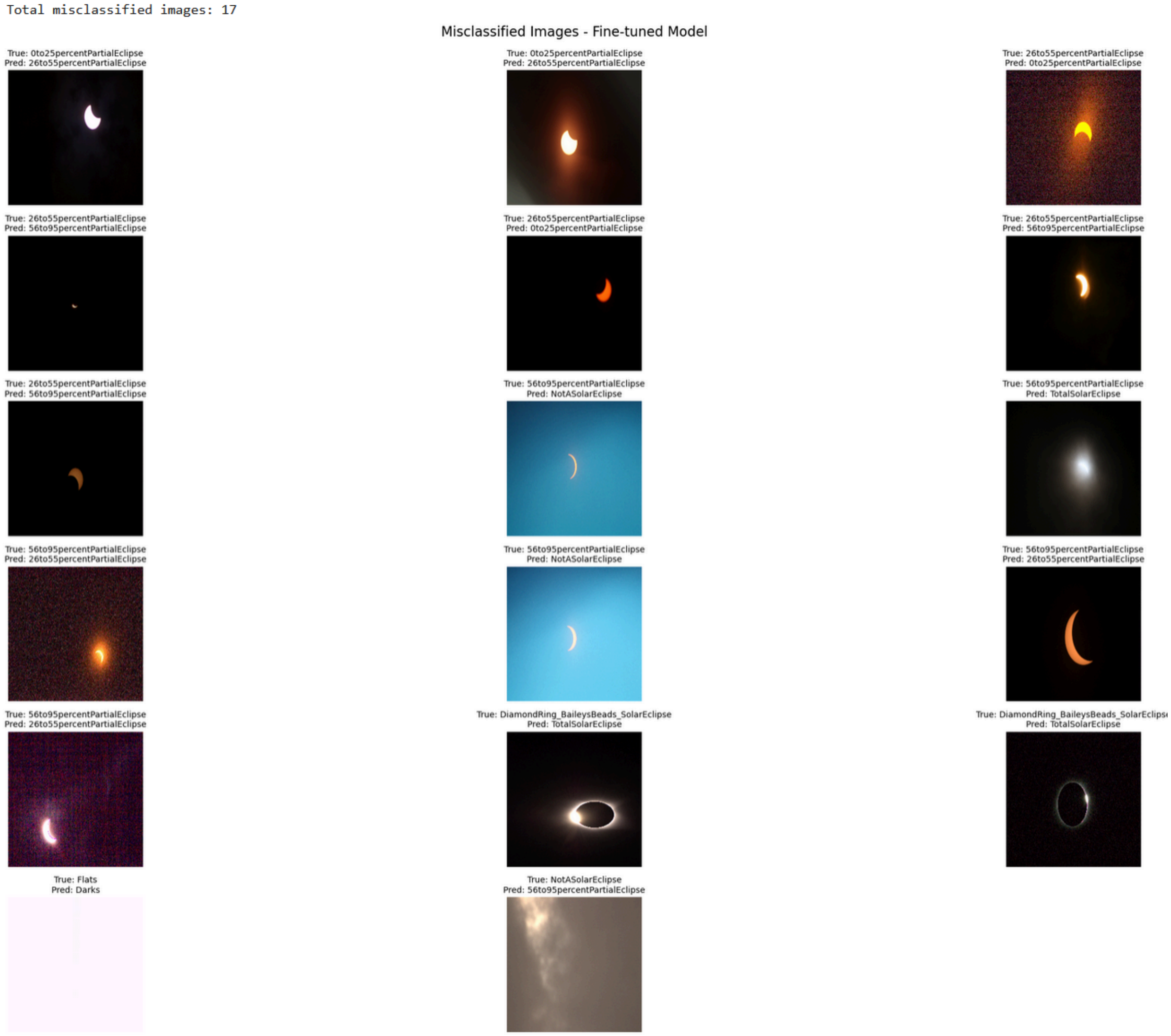
```
Fine-tuned Custom Model Accuracy: 0.8786
Logistic Regression Accuracy: 0.8000
SVM Accuracy: 0.7714
KNN Accuracy: 0.7143
```



	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
```

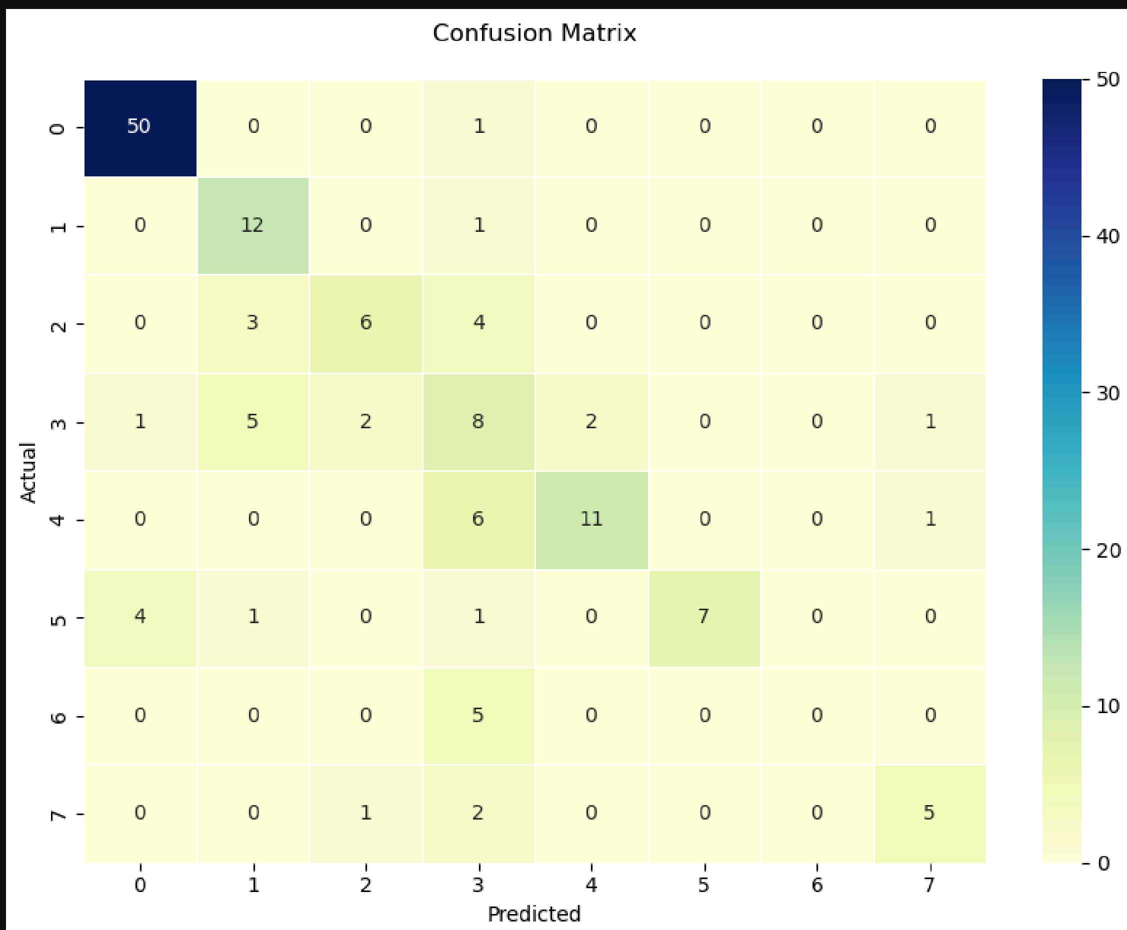


SIFT + BoVW + SVM/LR

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
```

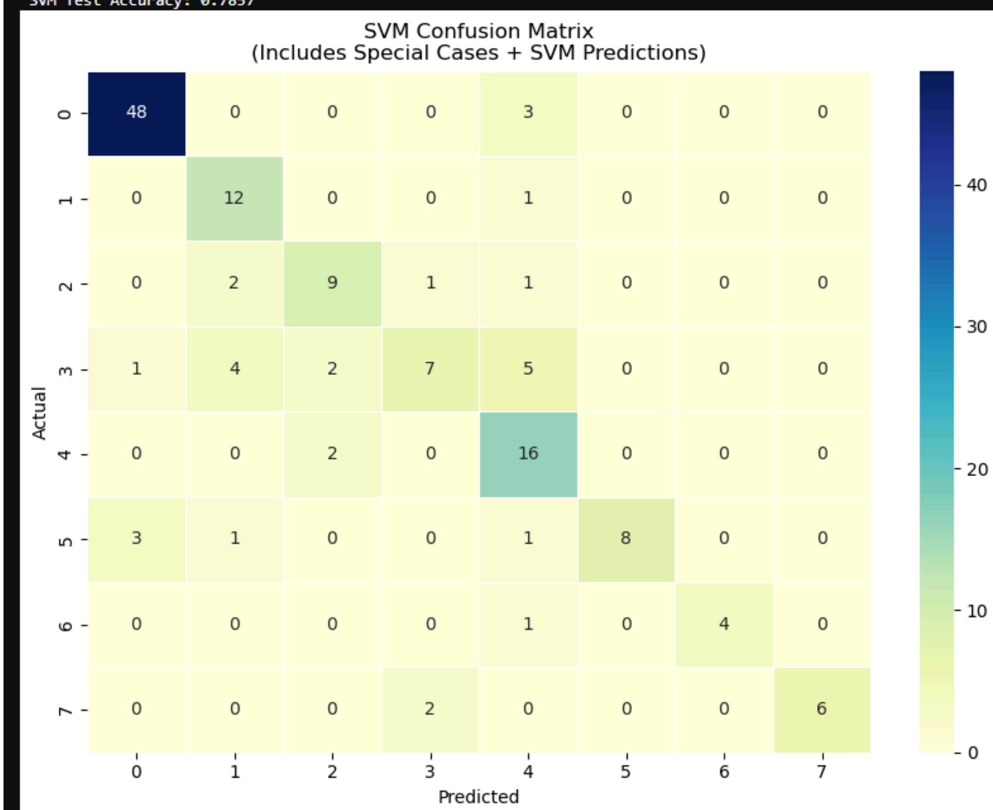



```
              precision    recall  f1-score   support

           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

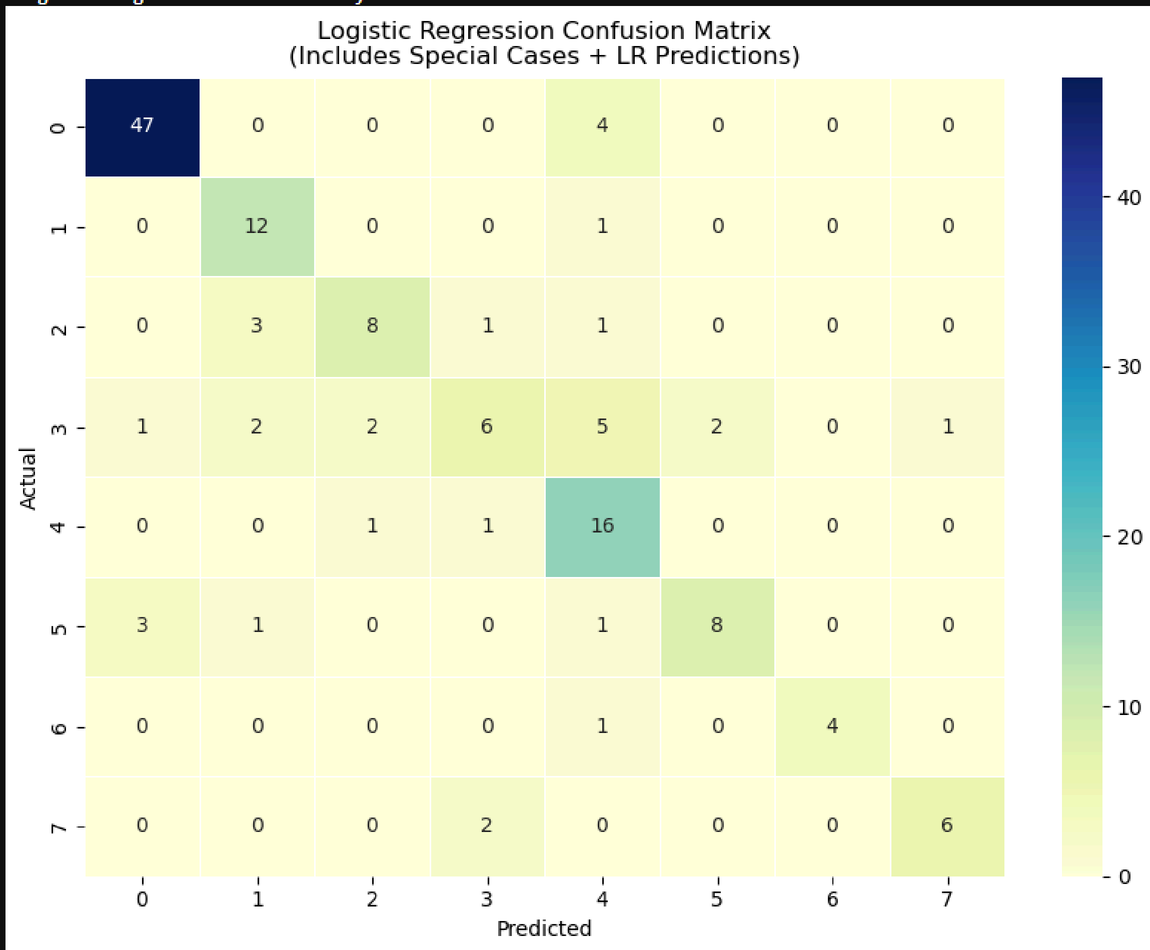
No matter what parameters we changed we could not get a higher accuracy. So we decided to look at what images were mostly being misclassified based on the confusion matrix and we can see that 6(flats) had no correct predictions and 4(darks) could do better. We decided that we could classify these two 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 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 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