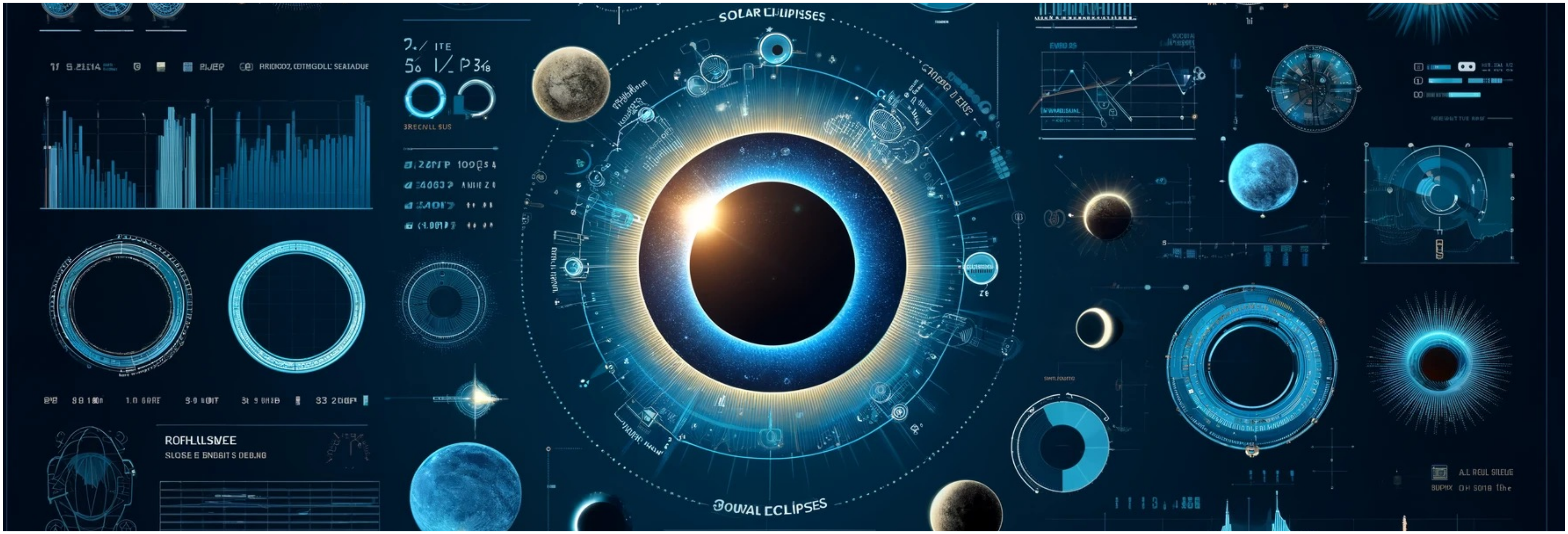


CATEGORIZING SOLAR ECLIPSE PHASES

FINAL PROJECT:

DEEP LEARNING – CONVOLUTIONAL NEURAL NETWORKS (CNN)

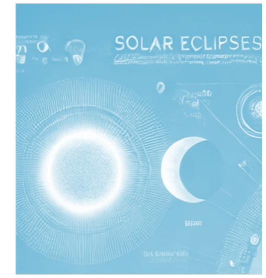
SOLAR ECLIPSE PHASES IMAGES CLASSIFICATION

A total solar eclipse happens when the Moon passes between the Sun and Earth, casting a shadow on Earth that fully blocks the Sun’s light in some areas.

This only happens occasionally, because the Moon doesn't orbit in the exact same plane as the Sun and Earth do. The time when they are aligned is known as eclipse season, which can happen twice a year.

This project is part of a Kaggle competition mission is to create the most accurate sorting machine that categorizes a solar eclipse photograph into a specific solar eclipse phase.

DATA LOADING, ANALYSIS AND CLEANING



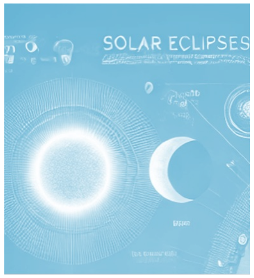
1. For this project, we will have access to a dataset of 495 pictures that were hand classified and 140 test pictures. In the train file metadata along with the picture ID we have the label category that indicates what kind of eclipse phase the picture corresponds.

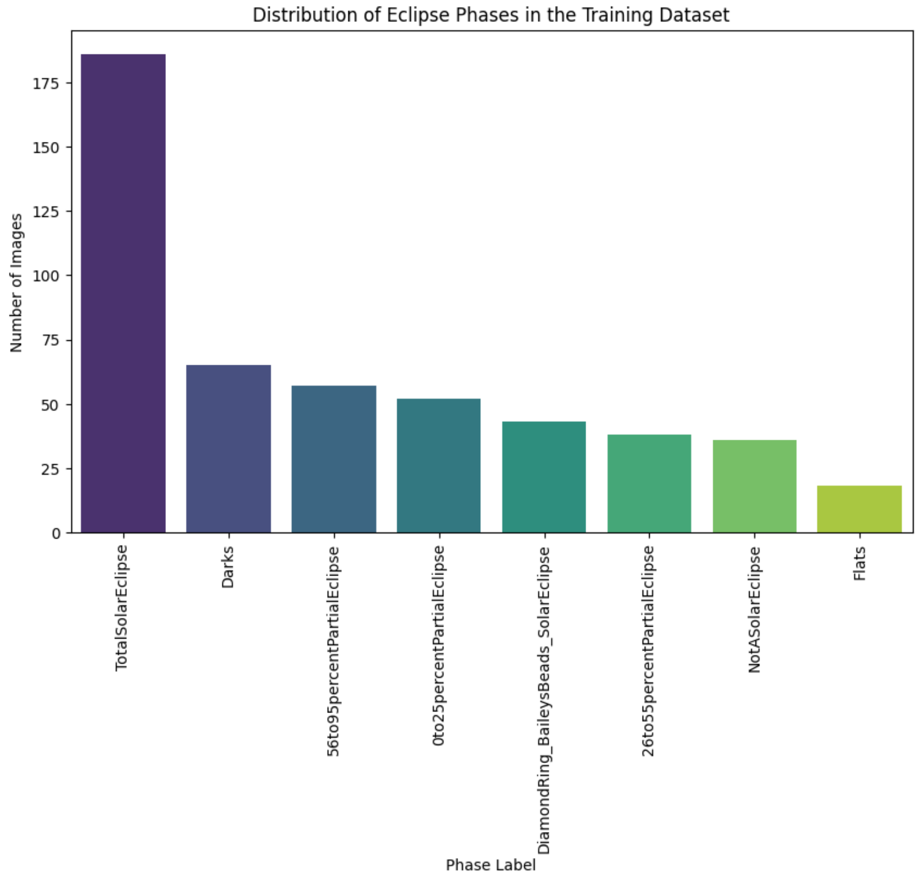
2. The following data sets are provided in the Kaggle page to train and test the model:
   - train - a folder containing the training images
   - test - a folder containing the test images
   - label_num_to_phase_map.json - The mapping between each phase code and the real phase name.
   - sample_submission.csv - a sample submission file in the correct format
   - train.csv - maps the training Image to the appropriate phase Id.

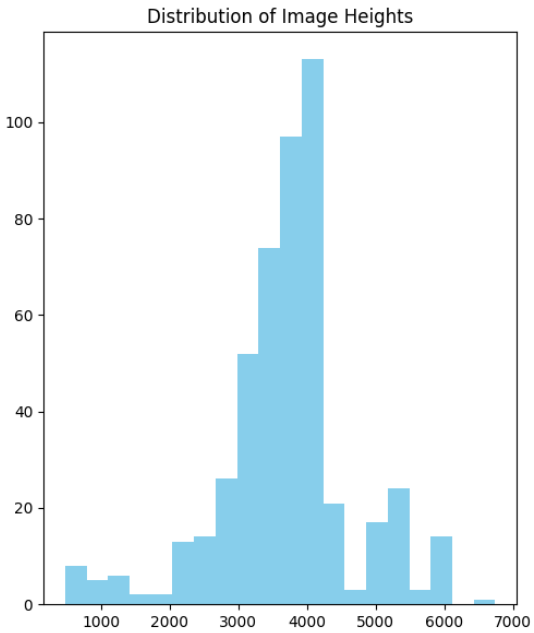
2. The data is complete and there is no irregularity, there is no need for any cleaning or manipulation. However the size and the resolution of the images is extremely heterogeneous.

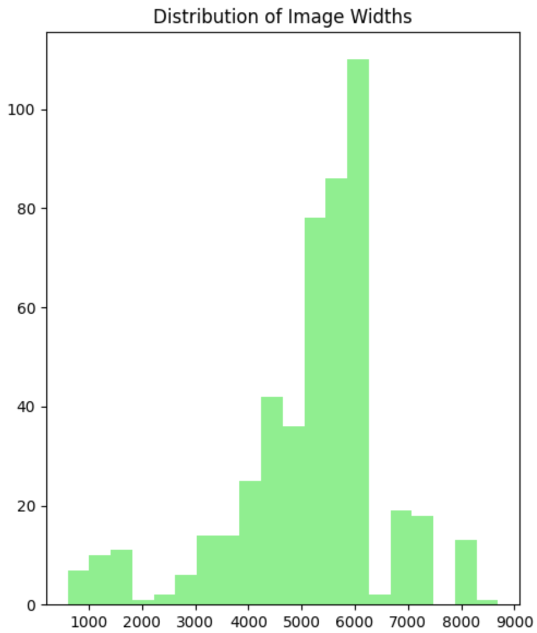
```
RangeIndex: 495 entries, 0 to 494
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   image_id    495 non-null    object
 1   label       495 non-null    int64
 2   phase_name  495 non-null    object
dtypes: int64(1), object(2)
memory usage: 11.7+ KB
```

VISUALIZATIONS - DISTRIBUTIONS

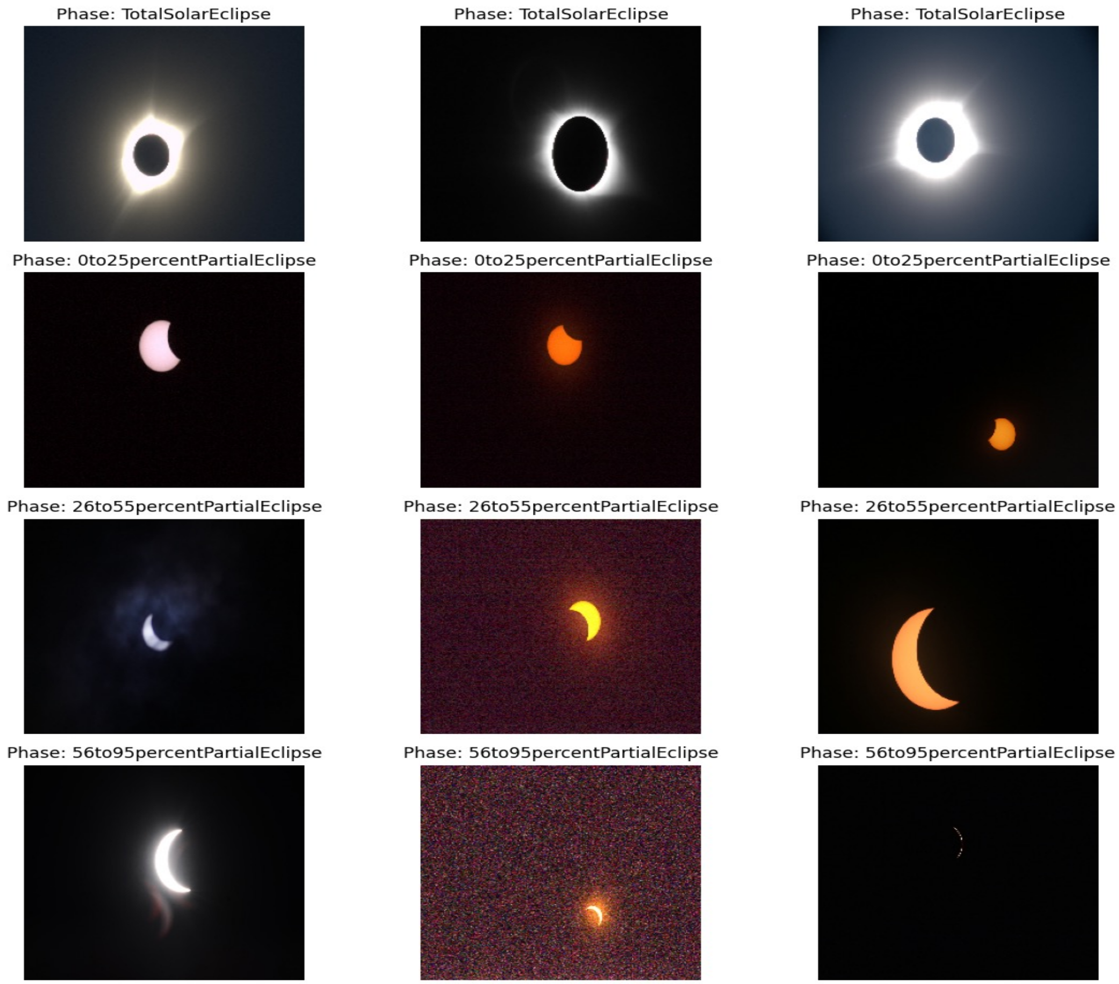


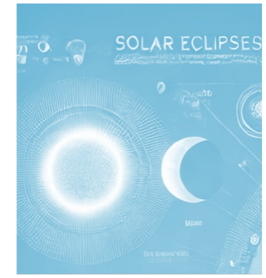




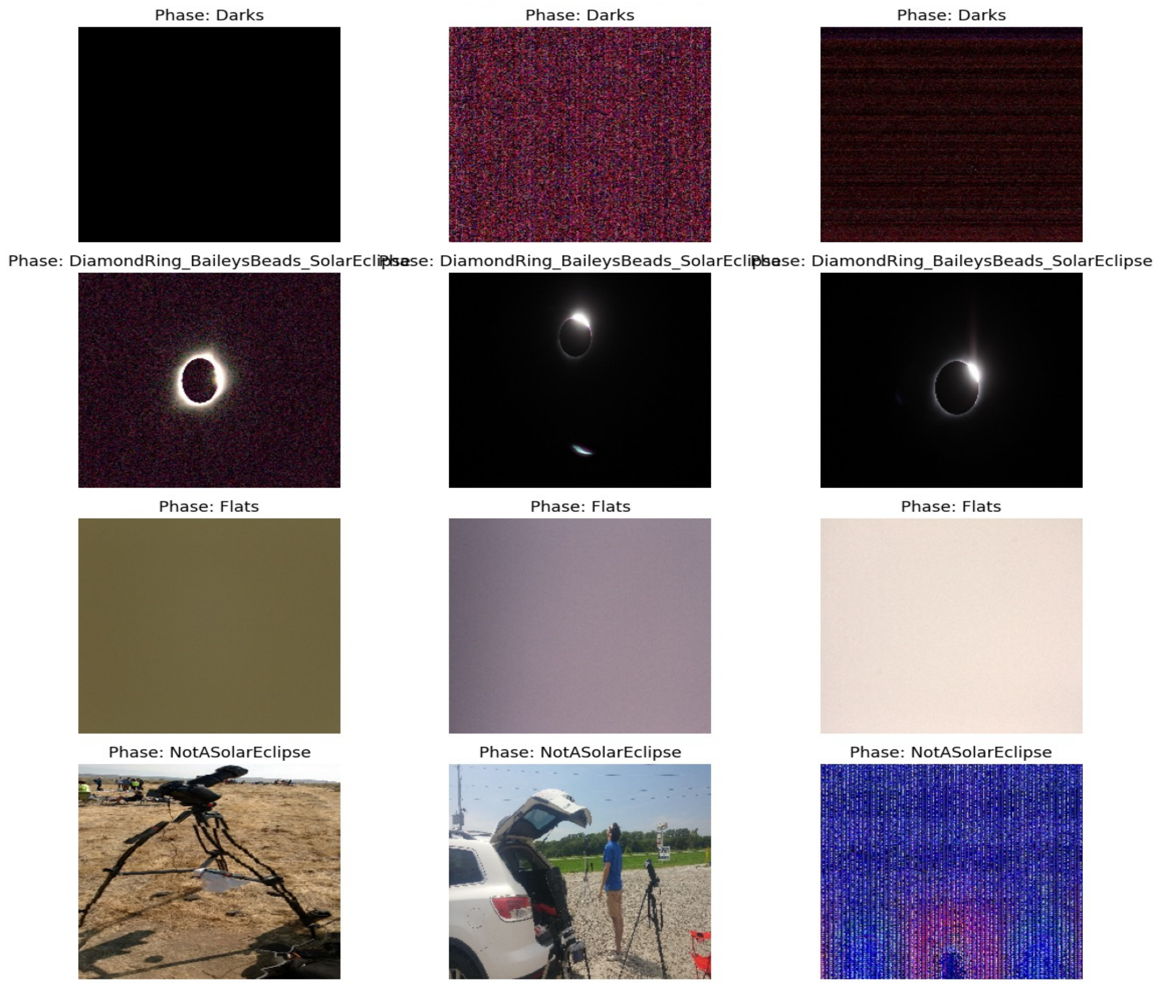


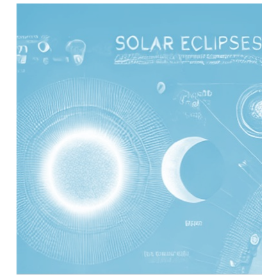
VISUALIZATIONS – IMAGE SAMPLES





VISUALIZATIONS – IMAGE SAMPLES





CNN MODELS FOR CLASSIFICATION

Model: "sequential_8"

Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
conv2d_16 (Conv2D)	(None, 7, 7, 128)	1,474,688
max_pooling2d_16 (MaxPooling2D)	(None, 3, 3, 128)	0
batch_normalization_16 (BatchNormalization)	(None, 3, 3, 128)	512
conv2d_17 (Conv2D)	(None, 3, 3, 256)	295,168
max_pooling2d_17 (MaxPooling2D)	(None, 1, 1, 256)	0
batch_normalization_17 (BatchNormalization)	(None, 1, 1, 256)	1,024
flatten_8 (Flatten)	(None, 256)	0
dense_24 (Dense)	(None, 512)	131,584
dropout_16 (Dropout)	(None, 512)	0
dense_25 (Dense)	(None, 256)	131,328
dropout_17 (Dropout)	(None, 256)	0
dense_26 (Dense)	(None, 8)	2,056

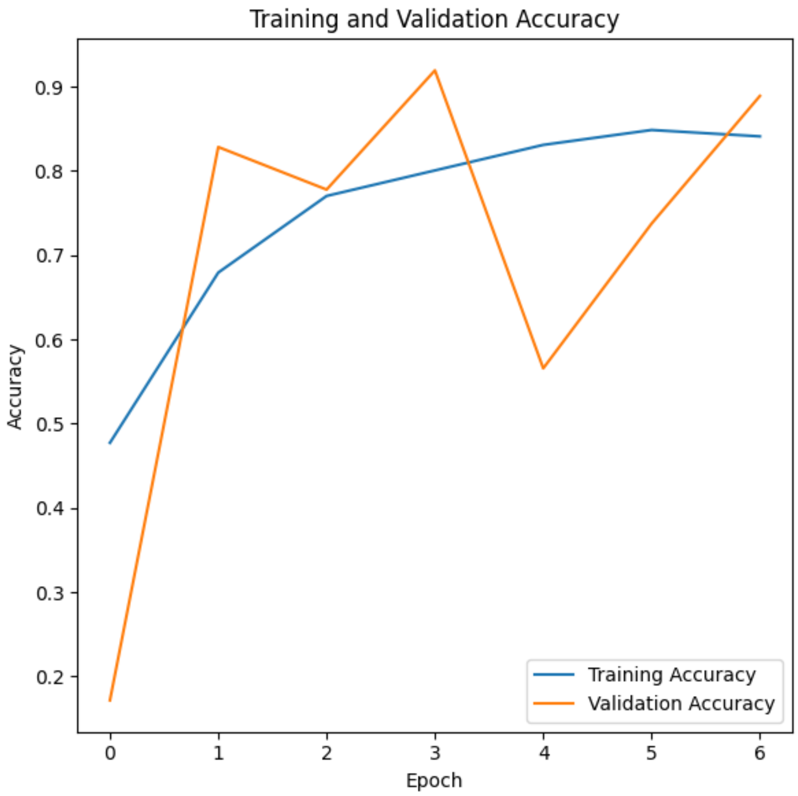
Total params: 8,365,530 (31.91 MB)

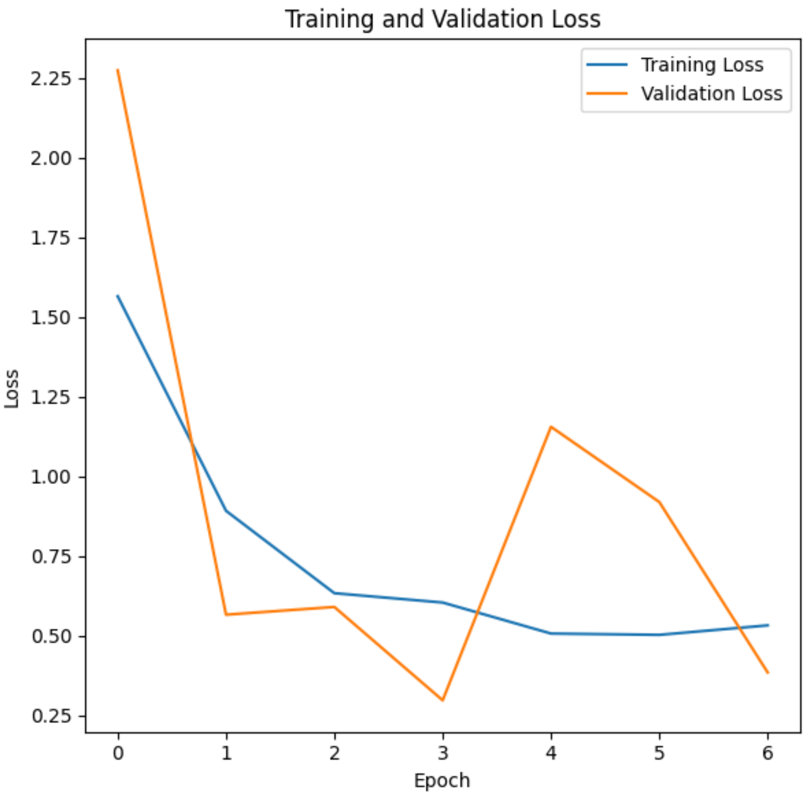
Trainable params: 2,035,592 (7.77 MB)

Non-trainable params: 2,258,752 (8.62 MB)

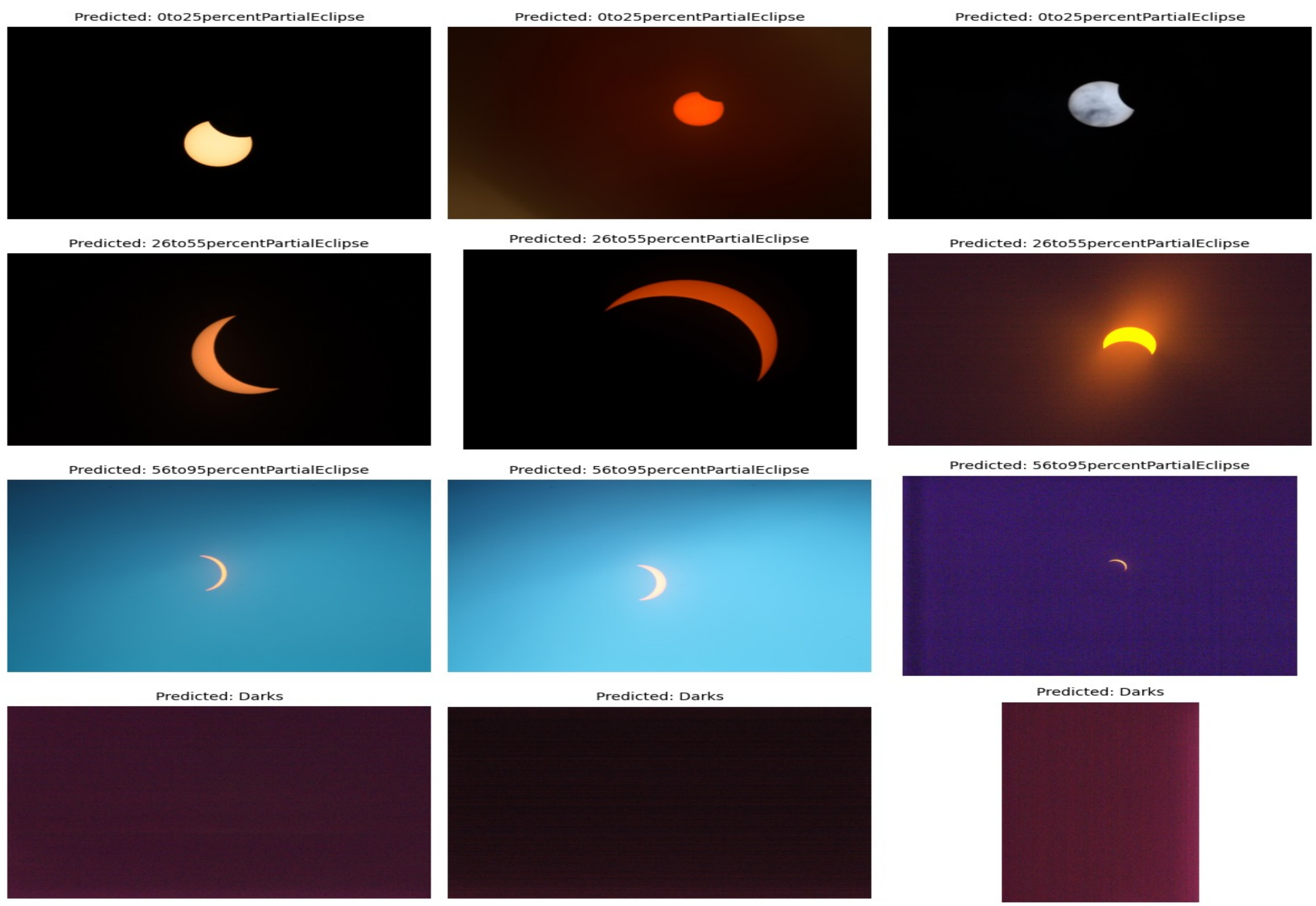
Optimizer params: 4,071,186 (15.53 MB)

MODEL PERFORMANCE

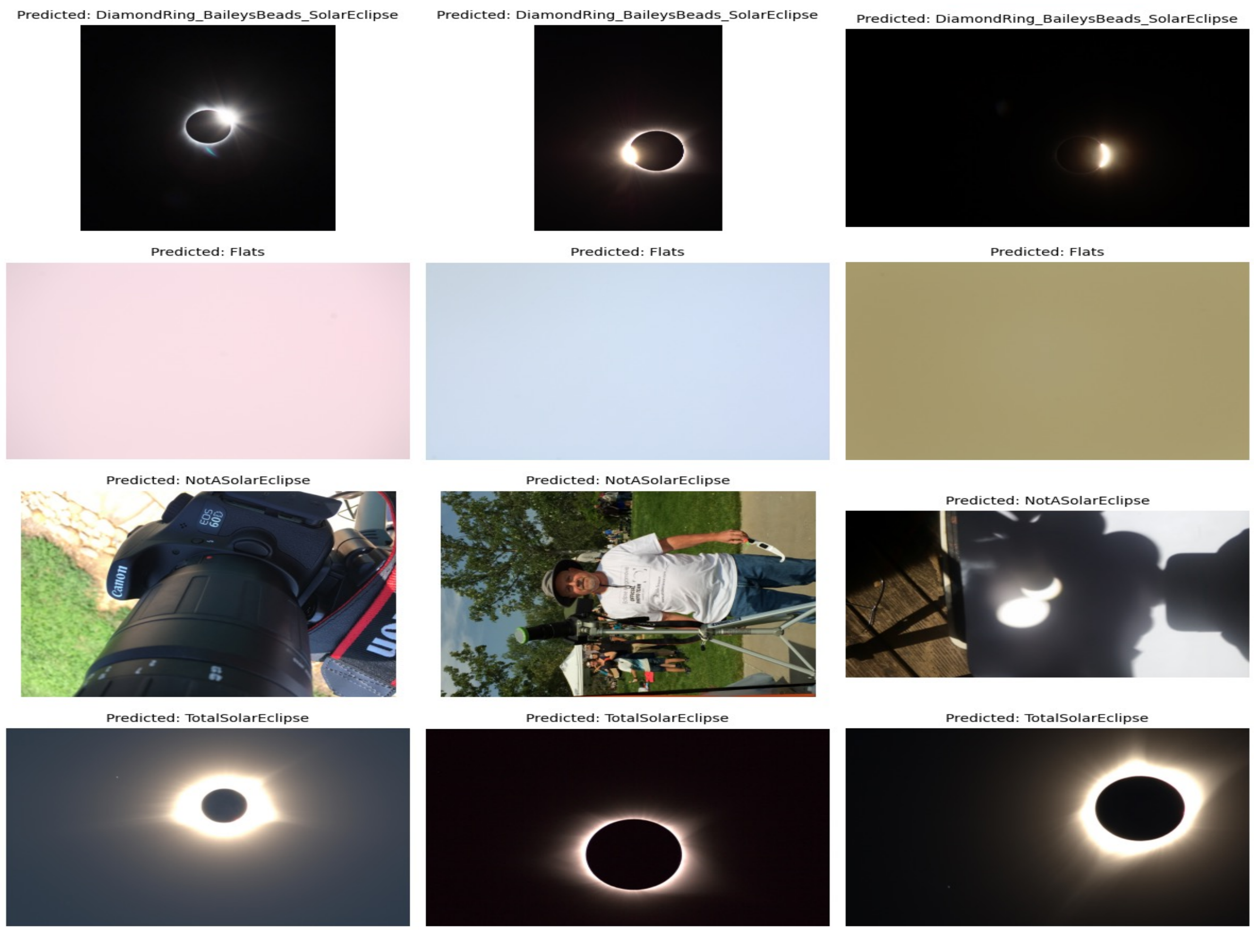


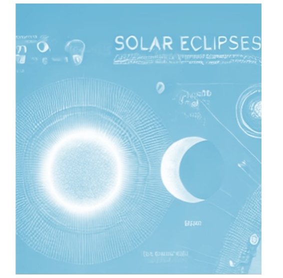


RESULTS SAMPLES



RESULTS SAMPLES





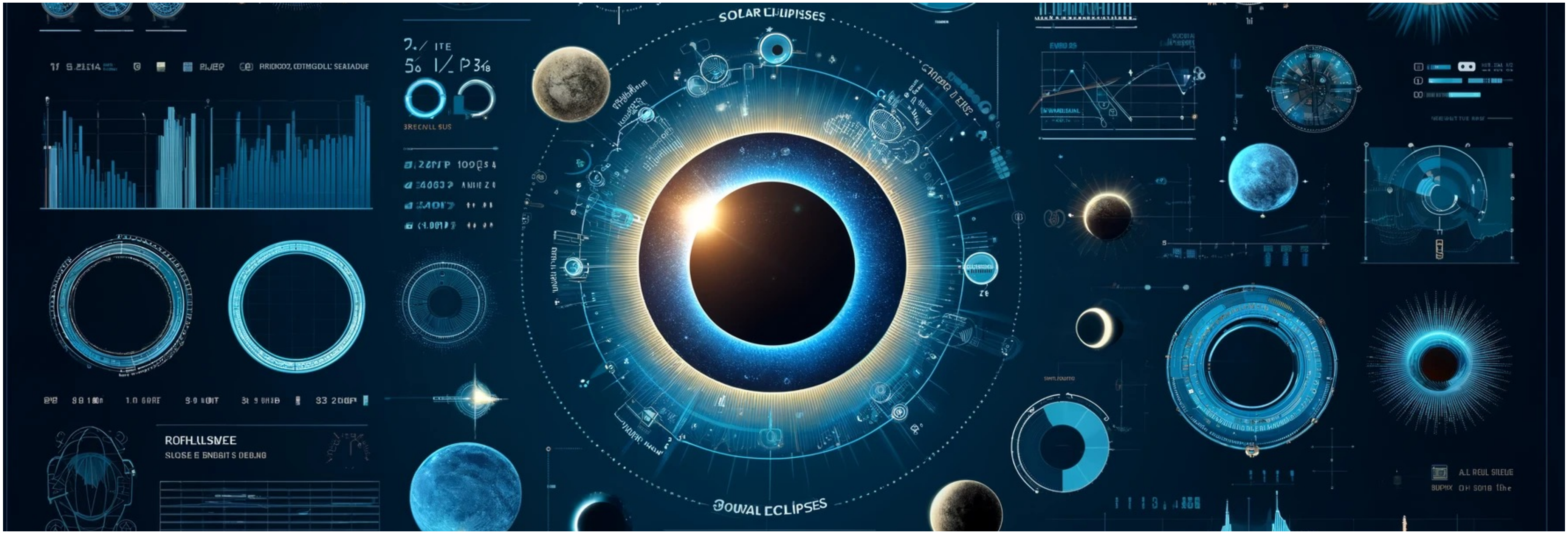
CONCLUSIONS

This project aimed to classify images of solar eclipses into different phases using a deep learning approach. The goal was to develop a model capable of accurately identifying the phase of an eclipse from a diverse set of images.

1. What Helped Improve Performance:
   - Using MobileNetV2 as a pre-trained model provided a solid foundation, significantly enhancing the model's ability to learn from the small dataset of 495 images.
   - Upscaling images to 224x224 allowed for better detail capture, which was crucial due to the heterogeneous nature of the image sizes and resolutions.
   - Hyperparameters tuning

2. What did not help
   - Adjusting the class imbalance by adding class weights to compensate for the overrepresented TotalSolarEclipse phase worsened the model’s performance.
   - Overfitting countermeasures :Early Stopping

3. Learnings and Takeaways
   - Key takeaways from this project are centered around the challenges posed by small, imbalanced datasets.
   - Overfitting and validation fluctuations are hard to avoid, especially with imbalanced data.
   - Data augmentation or synthetic data generation may be essential in future iterations to help balance the dataset.
   - Importance of careful hyperparameter tuning—from learning rates to dropout rates—in maintaining a balance between training speed, overfitting, and generalization performance



END

FINAL PROJECT:

DEEP LEARNING – CONVOLUTIONAL NEURAL NETWORKS (CNN)