# MOON PHASE DETECTION USING DEEP LEARNING

*Project report submitted in fulfilment of the requirements for the Degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

By

**JASWANTH SINGH KUMAR LANKADASU**

**12104841**

Supervisor

**MR. VED PRAKASH CHAUBEY**

**School Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

April,2024

# ABSTRACT

This report presents a novel approach to moon phase detection by harnessing the capabilities of deep learning. The study introduces a model that utilizes convolutional neural networks (CNNs) to analyse lunar images and accurately identify various moon phases. This method offers a significant improvement over traditional observational techniques, which are often limited by environmental conditions and human subjectivity.

The report outlines the development and validation of the deep learning model, emphasizing its robustness and adaptability to different imaging conditions. The model's performance is rigorously evaluated against a comprehensive dataset, demonstrating its precision and reliability in classifying moon phases.

The findings suggest that deep learning can play a crucial role in advancing astronomical studies, providing a scalable and objective tool for lunar observation. The research opens up possibilities for further applications of deep learning in astronomy, such as in the study of other celestial bodies and phenomena.

# DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled "MOON PHASE DETECTION USING DEEP LEARNING" in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr. Ved Prakash Chaubey. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

*Signature of Candidate*

**Jaswanth Singh Kumar Lankadasu**

**R.No: 12104841**

# SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the B.Tech Dissertation/dissertation proposal entitled **"MOON PHASE DETECTION USING DEEP LEARNING"**, submitted by **Jaswanth Singh Kumar Lankadasu** at **Lovely Professional University, Phagwara, India** is a bonafide record of his / her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

Signature of Supervisor

Ved Prakash Chaubey

**Date:**

**Counter Signed by:**

1) **Concerned HOD:**
   HoD's Signature: ________________

   HoD Name: ____________________

   Date: ___________________

2) **Neutral Examiners:**

   **External Examiner**

   Signature: _______________

   Name: __________________

   Affiliation: ______________

   Date: ___________________

   **Internal Examiner**

   Signature: _______________

   Name: __________________

   Date: ___________________

# ACKNOWLEDGEMENT

"GOD HELPS THOSE WHO HELP THEMSELVES."

"ARISE! AWAKE! AND STOP NOT UNTIL THE GOAL IS REACHED."

Success often requires preparation, hard work, and perspiration. The path to success is a long journey that calls for tremendous effort with many bitter and sweet experiences. This can only be achieved by the Graceful Blessing from the Almighty on everybody. I want to submit everything beneath the feet of God.

I want to acknowledge my regards to my teacher, Mr. Ved Prakash Chaubey, for his constant support and guidance throughout our project work. I would also like to thank, the academics team of School of Computer Science and Engineering for introducing such a great program.

I may be failing in my duties if I do not thank my parents for their constant support, suggestion, inspiration and encouragement and best wishes for our success. I am thankful for their supreme sacrifice, eternal benediction, and ocean-like bowls full of love and affection.

# TABLE OF CONTENTS

# TABLE OF CONTENTS

# List of Figures

# CHAPTER 1: INTRODUCTION

The celestial dance of the moon around Earth has captivated human imagination since time immemorial. Its cyclical phases, from the slender crescent to the full moon's luminous orb, have been a source of wonder, a marker of time, and a beacon for navigators. The study of these phases, known as moon phase detection, is not merely an exercise in astronomy; it is a thread woven into the fabric of our cultural and scientific heritage.

## 1.1 Importance of Moon Phase Detection

Moon phase detection holds paramount importance across various disciplines. In navigation, the moon's phases provide essential clues for celestial navigation, guiding sailors across the open seas. In agriculture, the lunar cycle influences planting and harvesting schedules, a practice that dates back to ancient civilizations. In science, understanding the moon's phases is crucial for space exploration, allowing for precise calculations in spacecraft trajectories and lunar landings. Moreover, the moon's phases have profound implications for Earth's tides, influencing marine and coastal ecosystems.

## 1.2 Challenges in Traditional Methods

Traditional methods of moon phase detection often rely on direct observation, which, while effective, come with limitations. Cloud cover, light pollution, and geographic constraints can obscure the moon's visibility, leading to inaccurate or missed observations. Additionally, manual tracking and prediction of moon phases are labor-intensive and prone to human error, making it challenging to maintain consistent and precise records over extended periods.

## 1.3 Advancements through Deep Learning

The advent of deep learning has ushered in a new era for moon phase detection. By leveraging complex algorithms and neural networks, deep learning models can analyse vast amounts of astronomical data with unprecedented accuracy and efficiency. These models can detect subtle patterns in lunar imagery, predict phases with high precision, and even compensate for visual obstructions such as clouds or shadows. The automation and scalability of deep learning have the potential to revolutionize how we observe and understand the moon, opening new frontiers in both practical applications and scientific inquiry.

As we delve deeper into the intricacies of moon phase detection through deep learning, we stand on the cusp of a paradigm shift. This chapter lays the groundwork for exploring the transformative impact of deep learning on this ancient practice, setting the stage for a detailed examination of its challenges and triumphs in the chapters to come. The moon's timeless journey across the night sky continues to inspire and inform us, and with the tools of modern technology, we are poised to unlock even more of its mysteries.

## 1.4 Theoretical Background

Automated moon phase detection utilizes a powerful combination of technologies: artificial intelligence (AI), machine learning (ML), deep learning (DL), and convolutional neural networks (CNNs). Understanding the relationships between these concepts is crucial for appreciating the approach taken in this report.

### 1.4.1 Artificial Intelligence (AI)

Artificial intelligence, in its broadest sense, refers to the field of computer science dedicated to creating intelligent machines capable of mimicking human cognitive functions such as learning, problem-solving, and decision-making. AI encompasses a wide range of techniques and approaches, with machine learning being a prominent subset.

### 1.4.2 Machine Learning (ML)

Machine learning algorithms are a type of AI that allows computers to learn from data without explicit programming. They can identify patterns and relationships within data, enabling them to make predictions or classifications on new, unseen data. Machine learning algorithms are trained on large datasets, during which they adjust internal parameters to improve their performance on a specific task.

### 1.4.3 Deep Learning (DL)

Deep learning is a subfield of machine learning inspired by the structure and function of the human brain. Deep learning algorithms, often referred to as artificial neural networks, are comprised of multiple interconnected layers that process information in a hierarchical fashion. These layers progressively extract higher-level features from the data, allowing the network to learn increasingly complex representations. Deep learning excels at tasks like image recognition and natural language processing, where vast amounts of data are involved.

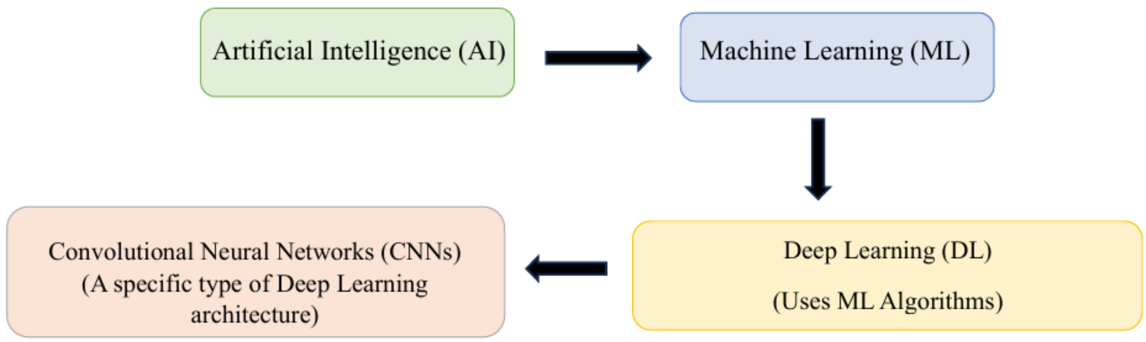


**Fig1: The AI Paradigm**

## CHAPTER 2: REVIEW OF LITERATURE

**[1]** This paper presents a compelling case for the dilated CNN model, which ingeniously replaces traditional convolution kernels with dilated counterparts, thereby expanding the receptive field without the burden of additional weights.

The study meticulously tests the proposed model on the Mnist handwritten digital recognition dataset, revealing a significant reduction in training time and a commendable increase in training accuracy. The hybrid dilated CNN (HDC) model further refines this approach by stacking dilated convolution kernels of varying dilation rates, addressing the detail loss issue prevalent in the dilated CNN model. The HDC's prowess is demonstrated on a wide-band remote sensing image dataset of Earth's terrain, showcasing a remarkable improvement in both training and testing accuracy.

**[2]** This research paper delves into the development of the SAZ algorithm, aiming to expedite the acquisition of lunar phase data. The study also evaluates the Raspberry Pi's capabilities as a compact and powerful CPU for a portable system designed to detect lunar phases. The SAZ algorithm, when combined with established techniques such as HSV, Canny, erosion, shape detection, and binarization, is rigorously tested on both personal computers and Raspberry Pi platforms.

The findings reveal that the SAZ algorithm significantly enhances the shape detection process, enabling the accurate revelation of moon phases. Moreover, the Raspberry Pi emerges as a viable CPU for a portable remote sensing structure, capable of performing lunar phase detection in a compact form factor.

**[3]** The cosmos, a vast expanse of mysteries and wonders, has always beckoned humanity to unravel its secrets. The field of astronomy, at the intersection of mathematics, physics, and chemistry, endeavours to decode the universe's origin, evolution, and the celestial bodies within. Artificial intelligence (AI) and deep learning have become pivotal in this quest, offering tools to parse through the astronomical data deluge and extract meaningful insights.

This review chronicles the evolution of connectionism in astronomy, tracing its journey from the early adoption of multilayer perceptrons to the sophisticated use of convolutional neural networks.

**[4]** This paper delves into the application of AI in astronomy, providing a concise introduction to machine learning and deep learning methods. It then illustrates their practical use through the examples of star-galaxy classification and the classification of low-mass X-ray binaries. The latter involves distinguishing between binaries hosting a neutron star and those with a black hole, a task that has been greatly enhanced by AI.

# CHAPTER 3: THE WORK

## 3.1 Problem Formulation

The problem at hand is the accurate detection of moon phases using deep learning techniques. Traditional methods, while rich in history, face challenges such as dependency on clear skies and manual interpretation. The objective is to formulate a deep learning model that can reliably predict moon phases from various image sources, overcoming the limitations of visibility and human error.

## 3.2 Objectives of the Study

The study aims to achieve several key objectives:

- Develop a deep learning model that can accurately classify the phases of the moon.
- Ensure the model's robustness against common issues like cloud cover and image distortion.
- Validate the model's performance against existing astronomical data and traditional observation methods.
- Explore the model's potential applications in other areas of astronomy and related fields.

## 3.3 Research Methodology

The core objective of this project lies in harnessing the power of deep learning, specifically Convolutional Neural Networks (CNNs), to accurately detect the various phases of the moon. To achieve this feat, a meticulous methodology was adopted, encompassing several key stages:

### 3.3.1. Data Acquisition and Preprocessing

The foundation of any deep learning project rests upon the data itself. In this instance, our journey begins with acquiring a comprehensive dataset of lunar images encompassing the entire spectrum of moon phases. Here, we leverage the Python programming language and its rich ecosystem of libraries. Essential libraries employed include:

- **OS**: This library provides functionalities for interacting with the operating system, enabling us to navigate to the directory containing the moon image dataset.
- **NumPy**: This library serves as the workhorse for numerical computations, facilitating the manipulation and storage of image data as multidimensional arrays.
- **TensorFlow**: As a leading open-source deep learning framework, TensorFlow provides the underlying infrastructure for building and training our CNN model.
- **Keras**: Sitting atop TensorFlow, Keras offers a high-level API, streamlining the creation and configuration of neural network architectures.

Once the dataset is secured, several crucial steps are undertaken to prepare it for analysis by the CNN model:

- **Image Count**: To gain a foundational understanding of the dataset's size and scope, we employ Python's functionalities to enumerate the total number of images present. This information proves valuable in determining the complexity of the model that can be effectively trained with the available data.

```
first quarter Folder has 105 Moon
full moon Folder has 213 Moon
new moon Folder has 25 Moon
third quarter Folder has 112 Moon
waning crescent Folder has 123 Moon
waning gibbous Folder has 99 Moon
waxing crescent Folder has 148 Moon
waxing gibbous Folder has 131 Moon
Moon Folder has 956 Images
```

**Fig2: An Overview of Dataset**

- **Image Loading**: The core images from the dataset are meticulously loaded into memory using libraries like NumPy. This process transforms the image files from their raw format into numerical representations amenable to manipulation by the CNN model.

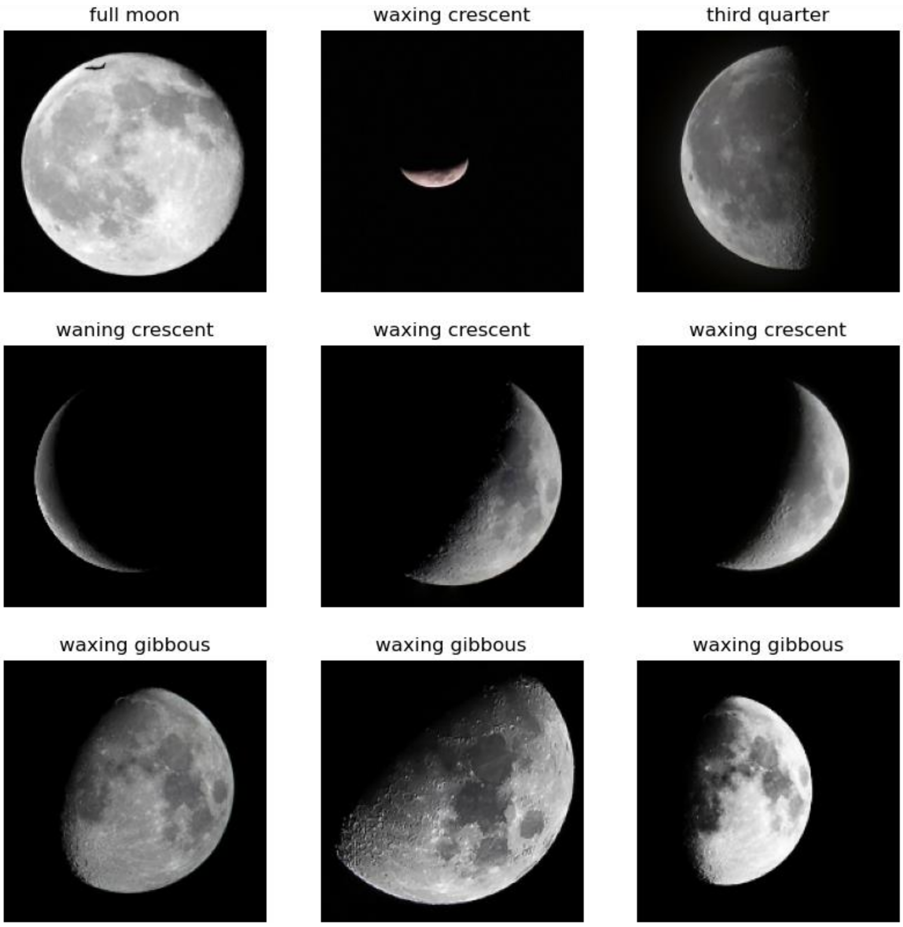


**Fig3: Few Moon Images from the dataset**

- **Dataset Splitting**: A critical step in deep learning involves splitting the dataset into two distinct subsets – the training set and the validation set. The training set constitutes the bulk of the data and serves as the training ground for the CNN model. As the model learns from these images, it establishes the intricate relationships between the pixel values and the corresponding moon phases. The validation set, typically amounting to a smaller portion of the data, plays a vital role in evaluating the model's performance during the training process. By exposing the model to unseen data from the validation set, we can assess its ability to generalize and identify moon phases on images it hasn't encountered before. Techniques like stratified sampling can be employed here to ensure both training and validation sets maintain a representative distribution of all moon phases.
- **Class Extraction**: Since our objective is to classify moon images into distinct phases (e.g., new moon, crescent, full moon), it's crucial to extract this classification information from the dataset. This typically involves associating each image with a numerical label corresponding to its specific moon phase.

```
['first quarter',
 'full moon',
 'new moon',
 'third quarter',
 'waning crescent',
 'waning gibbous',
 'waxing crescent',
 'waxing gibbous']
```

**Fig4: The various moon phase classes**

To gain a deeper understanding of the data and identify any potential issues, data visualization techniques can be employed at this stage. Libraries like Matplotlib and Seaborn can be used to generate histograms depicting the distribution of image pixels across different channels (e.g., RGB) and visualize sample images from each moon phase class.

### 3.3.2. Data Augmentation

While a comprehensive dataset is essential, it's equally important to account for the inherent variability present in real-world scenarios. Moon images captured under varying lighting conditions, camera angles, and atmospheric distortions can pose challenges for the CNN model. To address this, a technique known as data augmentation is employed. This process involves artificially manipulating the existing images in the training set to create new variations. The goal is to enrich the dataset with a wider range of examples, thereby enhancing the model's ability to recognize moon phases under diverse conditions. Common data augmentation techniques employed in this project include:

- **Random Flip (Vertical)**: This technique randomly flips a certain percentage of images vertically (upside down). This introduces variability that mimics the effect of the moon being captured from different perspectives and helps the model become less reliant on the specific orientation of the moon in the image.

- **Random Rotation (0.1)**: Images in the training set are subjected to random rotations of a small degree (e.g., +/- 0.1 radians). This simulates the subtle variations in the moon's apparent position in the night sky due to camera tilt or natural variations in perspective.
- **Random Zoom (0.1)**: A slight random zoom is applied to a subset of images in the training set (e.g., +/- 0.1 zoom factor). This technique mimics the effect of the moon appearing slightly closer or farther away in different images, depending on the focal length of the camera used to capture them.

Following the application of data augmentation techniques, it's beneficial to visualize the augmented images to assess the effectiveness of the chosen methods and ensure they produce realistic variations of the original moon images.

**Fig 5: Augmented Moon Images**

### 3.3.3 The Moon Phase Classifier: A Convolutional Neural Network

Having meticulously prepared the lunar image dataset, we embark on the creation of the CNN model, the heart of our moon phase detection system. This model will be trained to identify the distinct patterns and features within the images that correspond to each moon phase. We leverage Keras, the high-level API atop TensorFlow, to construct a sequential model, a common architecture for image classification tasks. The model comprises a series of stacked layers, each performing a specific transformation on the input data:

- **Data Augmentation Layer**: This layer incorporates the data augmentation techniques discussed previously (random flip, rotation, zoom) during the training process. By applying these transformations on the fly, the model is exposed to a wider variety of moon image variations, enhancing its generalizability.
- **Rescaling Layer**: Image pixel values often range between 0 and 255. To standardize the input data and facilitate the training process, this layer rescales the pixel values to a range between 0 and 1. This normalization step ensures all features within the image contribute equally during training.

**Core Convolutional Layers**:

The subsequent layers within the model are the essence of the CNN architecture:

- **Convolutional 2D Layer (3x)**: Multiple convolutional layers are employed in the model. These layers apply learnable filters (kernels) that slide across the width and height of the input image. As the filters move, they perform element-wise multiplication with the underlying image data, capturing specific features like edges, lines, and shapes. This process results in the creation of feature maps, which essentially represent the model's understanding of the essential characteristics within the image. The model typically employs multiple convolutional layers stacked on top of each other, with each layer progressively extracting higher-level features from the previous layer's outputs. In our case, we utilize three convolutional 2D layers to progressively extract increasingly complex features that differentiate between the various moon phases.

**Pooling Layers**:

Following each convolutional layer, a pooling layer is often incorporated. Pooling layers serve the purpose of down sampling the feature maps, reducing their dimensionality while retaining essential information. This not only improves computational efficiency but also helps to control overfitting, a phenomenon where the model becomes overly attuned to the training data and performs poorly on unseen examples. Common pooling techniques include max pooling, which selects the maximum value from a predefined window within the feature map, and average pooling, which computes the average value within the window.

- **Max Pooling 2D Layer**: In this project, we utilize max pooling 2D layers placed after each convolutional layer. These layers reduce the dimensionality of the feature maps by selecting the maximum pixel value within a specific window size (e.g., 2x2). This process retains the most prominent features while discarding less significant details.

**Regularization with Dropout**:

- **Dropout Layer**: Dropout is a powerful technique employed to prevent overfitting. During training, a random subset of neurons within a layer is temporarily disabled (dropped out) at each training iteration. This forces the model to learn robust features that are not overly reliant on any specific neuron within a layer. The dropout rate, typically a value between 0 and 1, determines the proportion of neurons to be dropped out.

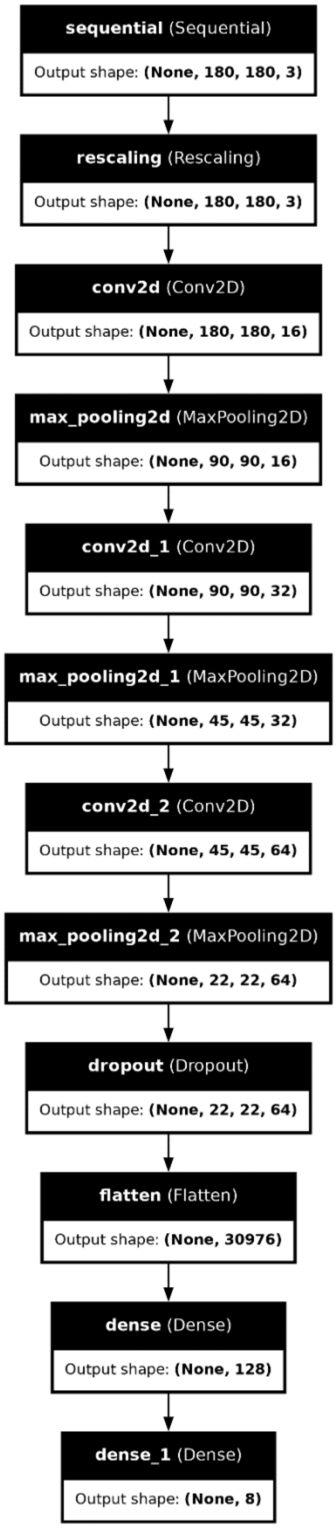


**Fig 6: The CNN Architecture**

**Classification Layers**:

The final stages of the model architecture focus on classifying the extracted features into distinct moon phases:

- **Flatten Layer**: Prior to feeding the extracted features into the classification layers, the multidimensional feature maps need to be flattened into a single-dimensional vector. This transformation prepares the data for compatibility with the fully connected layers that follow.
- **Dense Layer**: A dense layer, also known as a fully connected layer, is a core component of many neural networks. Unlike convolutional layers that operate on spatial data, dense layers perform matrix multiplications between the input vector and a weight matrix. This layer learns to combine the extracted features from the previous layers into a more abstract representation that is suitable for moon phase classification. Here, we employ one or more dense layers with a suitable number of neurons (activation units) to capture the complex relationships between the features and the corresponding moon phases.
- **Output Layer (Dense Layer with Softmax Activation)**: The final layer within the model serves as the classification layer. This layer typically utilizes a softmax activation function, which maps the output of the previous dense layer into a probability distribution across all possible moon phase classes (e.g., new moon, crescent, full moon). The softmax function ensures that the outputs of the layer sum to 1, allowing us to interpret the output as the probability of the input image belonging to each moon phase class.

### 3.3.4. Model Compilation and Training

With the CNN model meticulously crafted, we proceed to the training phase, where the model learns to identify moon phases from the prepared dataset. Here, several crucial steps are involved:

- **Optimizer**: An optimizer serves as the guiding force during the training process. It iteratively adjusts the weights and biases within the model's layers to minimize a predefined loss function. In this project, we can leverage an optimizer like Adam (Adaptive Moment Estimation), which has proven effective in various deep learning tasks. Adam incorporates momentum and adaptive learning rates, enabling the model to converge on optimal weights more efficiently.
- **Loss Function**: The loss function quantifies the discrepancy between the model's predictions and the actual moon phases represented by the training data. A commonly used loss function for multi-class classification problems like this is sparse categorical crossentropy. This function measures the average difference between the probability distribution generated by the model's output layer (softmax) and the one-hot encoded representation of the true moon phase for each image.
- **Metrics**: While the loss function guides the training process, it's equally important to evaluate the model's performance on unseen data. Metrics like accuracy, precision, recall, and F1-score are employed to assess the model's effectiveness in identifying moon phases. Accuracy represents the overall proportion of correctly classified images. Precision measures the ratio of true positives (correctly identified moon phases) to all positive predictions made by the model. Recall, on the other hand, focuses on the proportion of true positives out of all actual moon phases present in the data. F1-score

provides a harmonic mean of precision and recall, offering a balanced view of the model's performance.

**Training the Model**:

- **Epochs**: One training epoch refers to a single iteration where the model processes the entire training dataset once. During each epoch, the model calculates the loss for each image, backpropagates the error through the network, and updates the weights and biases based on the chosen optimizer. Training typically involves running the model for multiple epochs until it achieves an optimal level of performance.
- **Batch Size**: The training data is not fed into the model all at once. Instead, it's divided into smaller batches, which are processed sequentially during each training step. The batch size can significantly impact the training process. Smaller batch sizes lead to more frequent updates of the weights but can be computationally expensive. Conversely, larger batch sizes offer faster training but may result in slower convergence to the optimal solution.

**Monitoring and Early Stopping**:

- **Validation Set**: As mentioned earlier, the validation set plays a critical role in preventing overfitting. During training, the model's performance is evaluated on the validation set periodically. If the model's accuracy on the validation set starts to deteriorate while the training accuracy continues to improve, it signifies overfitting.
- **Early Stopping**: To address overfitting, a technique known as early stopping can be implemented. Here, the model's performance on the validation set is monitored. If the validation accuracy doesn't improve for a predefined number of consecutive epochs, the training process is halted. This prevents the model from memorizing the training data and ensures it generalizes well to unseen examples.

**Model: "sequential_1"**

| Layer (type) | Output Shape | Param # |
|---|---|---|
| sequential (Sequential) | ? | 0 |
| rescaling (Rescaling) | ? | 0 (unbuilt) |
| conv2d (Conv2D) | ? | 0 (unbuilt) |
| max_pooling2d (MaxPooling2D) | ? | 0 (unbuilt) |
| conv2d_1 (Conv2D) | ? | 0 (unbuilt) |
| max_pooling2d_1 (MaxPooling2D) | ? | 0 (unbuilt) |
| conv2d_2 (Conv2D) | ? | 0 (unbuilt) |
| max_pooling2d_2 (MaxPooling2D) | ? | 0 (unbuilt) |
| dropout (Dropout) | ? | 0 |
| flatten (Flatten) | ? | 0 (unbuilt) |
| dense (Dense) | ? | 0 (unbuilt) |
| dense_1 (Dense) | ? | 0 (unbuilt) |

**Total params:** 0 (0.00 B)

**Trainable params:** 0 (0.00 B)

**Non-trainable params:** 0 (0.00 B)

**Fig 7: The Model Summary**

### 3.3.5. Model Evaluation

Once the training process is complete, a comprehensive evaluation of the model's performance is undertaken. Here, we leverage the various metrics discussed earlier (accuracy, precision, recall, F1-score) to assess the model's effectiveness in classifying moon phases:

- **Classification Report**: A classification report summarizes the model's performance on the validation or test set, providing metrics like accuracy, precision, recall, and F1-score for each individual moon phase class. This detailed breakdown allows us to identify potential shortcomings in the model's ability to distinguish between specific moon phases.

**Visualization Techniques**:

- **Confusion Matrix**: A confusion matrix is a visual representation of the model's performance on a classification task. It depicts the number of correctly classified and misclassified images for each moon phase class. This visualization helps to identify moon phases that the model frequently confuses with others.
- **Loss and Accuracy Plots**: Plotting the training and validation loss (or accuracy) over epochs provides valuable insights into the training process. Ideally, the training loss should decrease steadily as the model learns, while the validation loss should either plateau or decrease slightly. Significant divergence between training and validation loss indicates potential overfitting.

**Manual Validation**:

In addition to the automated evaluation metrics, manually validating the model's performance can offer valuable insights. This involves feeding unseen moon images (not part of the training or validation sets) into the model and verifying its predictions. This step helps to assess the model's generalizability and identify any potential biases or limitations.

## FLOWCHART: THE MODEL

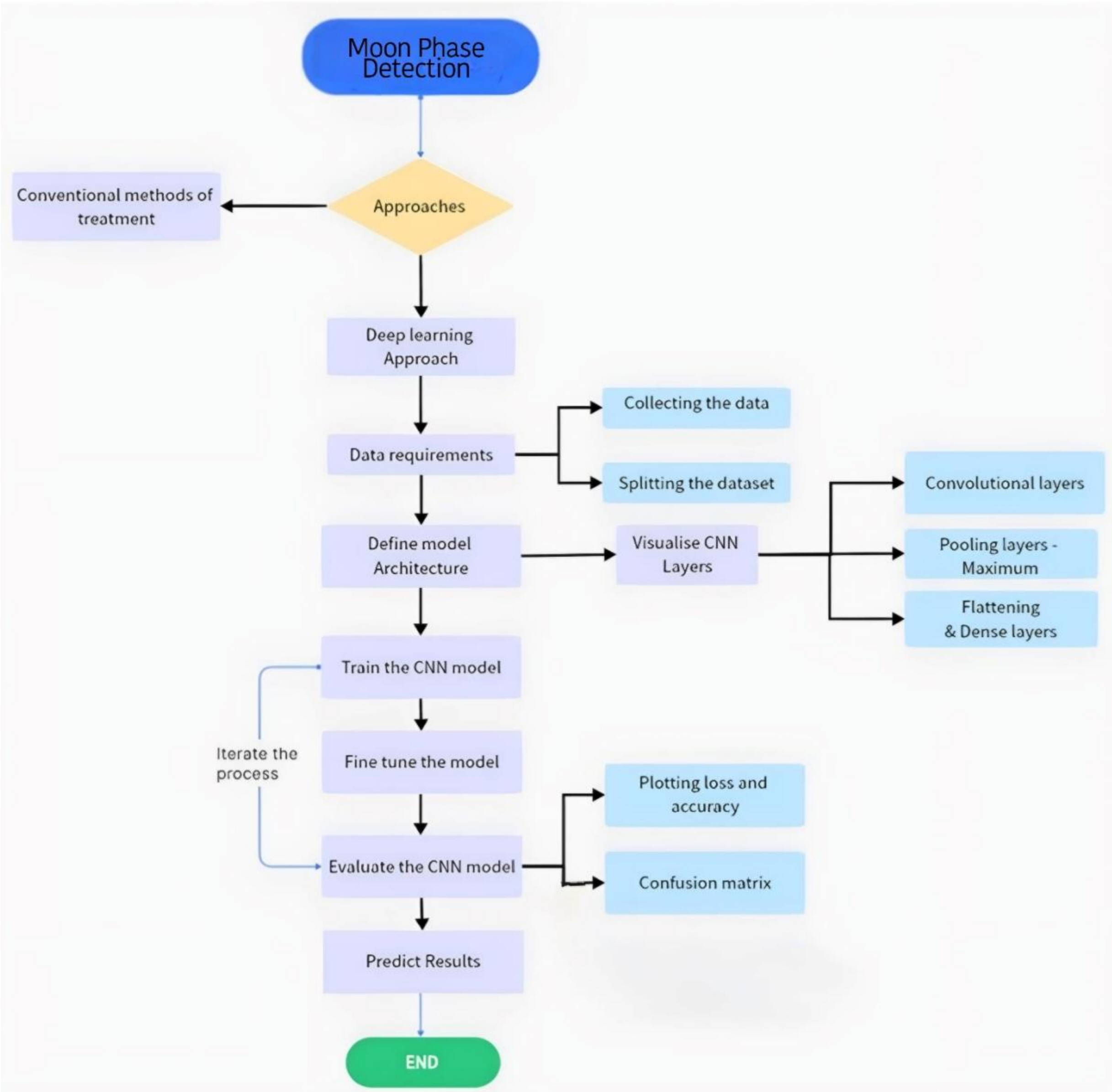


**Fig 8: The Project Flow**

.

## CHAPTER 4: RESULTS AND DISCUSSION

This section delves into the captivating results yielded by the model. We present the classification accuracy achieved, along with a breakdown of performance for each individual moon phase class. By analysing the various evaluation metrics, we gain valuable insights into the model's strengths and weaknesses.

### 4.1 The Results

#### Analysis of the Classification Report

The classification report provides a detailed evaluation of the model's performance on the test dataset. It presents several metrics for each moon phase class:

- **Precision**: This metric indicates the proportion of images classified as a specific moon phase that actually belong to that class. For instance, a precision of 0.86 for class 2 suggests that 86% of the images predicted as class 2 were truly moon phase 2.
- **Recall**: This metric represents the proportion of actual images belonging to a specific moon phase class that were correctly identified by the model. A recall of 0.72 for class 1 signifies that the model identified 72% of the actual class 1 images correctly.
- **F1-score**: This metric offers a harmonic mean between precision and recall, providing a balanced view of the model's performance for each class.

#### Overall Observations

The average accuracy of the model is 73%, indicating that it correctly classified 73% of the moon phase images in the test dataset. While this is a promising result, it's important to delve deeper into the performance of each individual class.

#### Classes with High Performance

Classes 2 (0.86 precision, 0.84 recall) and 7 (0.82 precision, 0.72 recall) exhibit the highest precision and recall values, signifying that the model is highly accurate in identifying these particular moon phases.

#### Classes with Improvement Potential

Classes 1 (0.60 precision, 0.72 recall), 3 (0.90 precision, 0.64 recall), and 6 (0.54 precision, 0.54 recall) show lower precision or recall scores. This suggests that the model might be misclassifying some images belonging to these classes or failing to identify all the relevant images. Further investigation into these classes might be necessary to understand the specific challenges faced by the model.

```
64/64 ━━━━━━━━━━━━━━━━━━━━ 2s 33ms/step - accuracy: 0.8314 - loss: 0.5023
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.7446 - loss: 0.7046
Training Accuracy: 0.8366013169288635
Validation Accuracy: 0.7329843044281006
```

**Fig 9: The Training and Validation Accuracy**

| Classification Report: | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.60 | 0.72 | 0.65 | 25 |
| 1 | 0.86 | 0.84 | 0.85 | 38 |
| 2 | 0.71 | 0.71 | 0.71 | 7 |
| 3 | 0.90 | 0.64 | 0.75 | 28 |
| 4 | 0.71 | 0.86 | 0.78 | 29 |
| 5 | 0.69 | 0.73 | 0.71 | 15 |
| 6 | 0.54 | 0.54 | 0.54 | 24 |
| 7 | 0.82 | 0.72 | 0.77 | 25 |
| accuracy | | | 0.73 | 191 |
| macro avg | 0.73 | 0.72 | 0.72 | 191 |
| weighted avg | 0.75 | 0.73 | 0.73 | 191 |

**Fig 10: The Classification Report**

### Accuracy and Loss Curves

The accuracy and loss curves visualize the model's performance during the training process. The x-axis represents the training epochs, and the y-axis represents the corresponding values for accuracy (left) and loss (right).

- **Training Accuracy**: The training accuracy curve depicts the model's performance on the training data with each epoch. Ideally, this curve should exhibit a monotonic increase, signifying the model's growing ability to correctly classify moon phases within the training set.
- **Validation Accuracy**: The validation accuracy curve represents the model's performance on the validation data after each epoch. This curve serves as a crucial indicator of generalization – the model's ability to perform well on unseen data. In an ideal scenario, the validation accuracy curve should also increase alongside the training accuracy, but at a potentially slower rate. A significant discrepancy between the two curves might suggest overfitting, where the model is memorizing the training data but failing to generalize to unseen examples.
- **Training Loss**: The training loss curve depicts the loss function calculated over the training data with each epoch. The loss function measures the discrepancy between the model's predictions and the actual moon phases. Ideally, the training loss curve should exhibit a monotonic decrease as the model learns to reduce its errors.
- **Validation Loss**: The validation loss curve represents the loss function calculated over the validation data with each epoch. Similar to the validation accuracy curve, this curve monitors the model's generalization ability. A continuous decrease in validation loss is desirable.

**Observations from the Curves**

- The training accuracy appears to increase steadily throughout the epochs, suggesting the model is effectively learning from the training data.
- The validation accuracy also seems to increase with each epoch, indicating that the model is generalizing well to unseen data in the validation set. The gap between the training and validation accuracy curves seems minimal, further supporting this observation.
- The training loss curve exhibits a downward trend, signifying a reduction in the model's errors over the training epochs.
- The validation loss curve also appears to decrease with each epoch, indicating that the model is learning to minimize its errors on unseen data as well.

Overall, the accuracy and loss curves suggest that the model has been successfully trained to identify moon phases. The increasing accuracy and decreasing loss on both the training and validation sets demonstrate the model's ability to learn and generalize effectively.

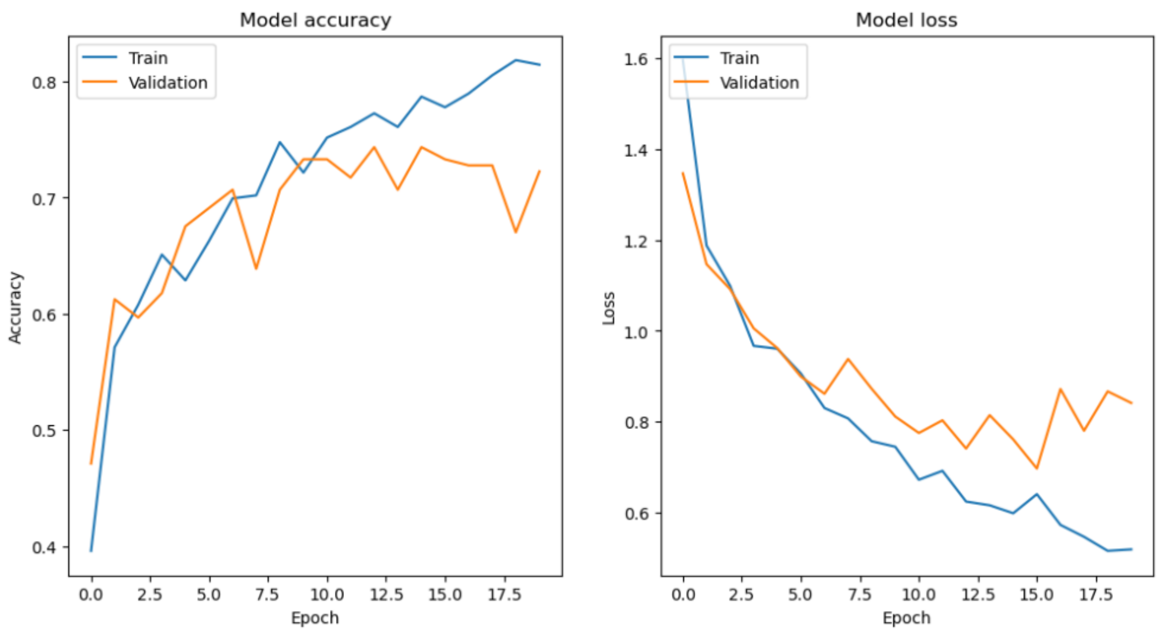


**Fig 11: The Accuracy and Loss Curves**

**The Confusion Matrix**

The confusion matrix, is a valuable tool for visualizing the performance of a classification model on a test dataset. It provides an insightful breakdown of how many images from each actual moon phase class were predicted by the model into each predicted moon phase class.

**Understanding the Confusion Matrix**

- **Rows**: The rows of the confusion matrix represent the actual moon phases present in the test dataset. For instance, the row labelled "First Quarter" indicates the number of images that were truly in the first quarter phase.

- **Columns**: The columns represent the moon phases that the model predicted for the images in the test dataset. The column labelled "First Quarter" corresponds to the number of images the model classified as first quarter, regardless of their actual phase.
- **Diagonal Values**: Ideally, the highest values should lie along the diagonal of the matrix. These values represent the number of images that were correctly classified by the model. For example, the value at the intersection of the "Full Moon" row and "Full Moon" column represents the number of images that were actual full moons and were also correctly predicted as full moons by the model.
- **Off-Diagonal Values**: Off-diagonal values represent misclassifications. A high value at a specific row (actual class) and a different column (predicted class) signifies that the model frequently misclassified images belonging to that actual class as the class indicated by the column.

**Observations from the Confusion Matrix**

- Examining the diagonal elements of the confusion matrix, we can see that most moon phases were classified correctly by the model to some extent.
- Class 1 (likely full moon) has a relatively high value (32) on the diagonal, indicating good performance in identifying full moons.
- Class 5 (waxing crescent) also shows a decent value (22) on the diagonal, suggesting the model accurately classified many waxing crescent images.
- However, there are also some off-diagonal elements with notable values, suggesting potential areas for improvement.
    - Class 3 (waning crescent) has a value of 18 in the "First Quarter" column, indicating that the model might be misclassifying some waning crescent images as first quarter.
    - Class 8 (full moon - waxing crescent) has a value of 11 in the "Waxing Crescent" column, suggesting some full moon images might be getting misclassified as waxing crescent.

Overall, the confusion matrix provides valuable insights into the model's performance for each moon phase class. While the model exhibits good accuracy for some classes, there are also instances of misclassification that warrant further investigation.

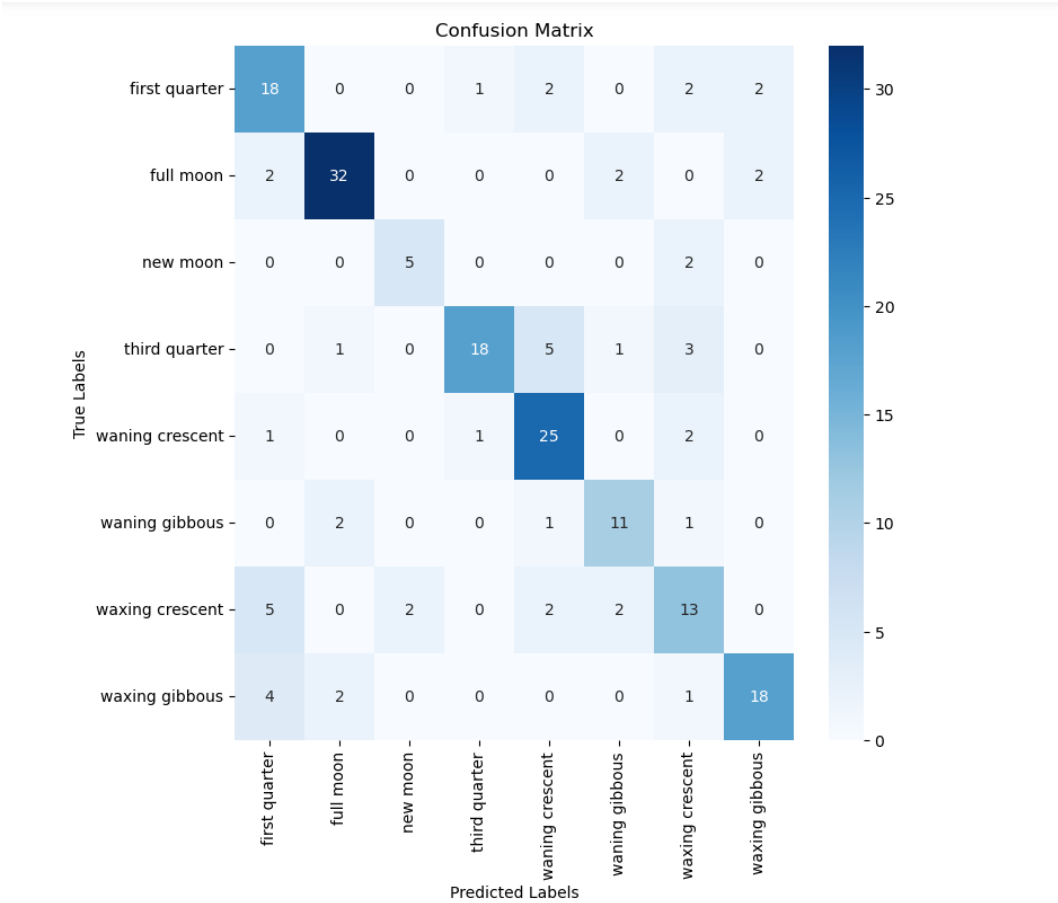


**Fig12: The Confusion Matrix**

## Manual Validation

Manual validation serves as a final layer of verification for the performance of the moon phase classification model. In this process, a set of unseen moon images (not part of the training or validation sets) is fed into the model, and its predictions are compared to human judgment. This step helps to assess the model's generalizability to real-world scenarios and identify any potential biases or limitations that might not have been revealed by the automated evaluation metrics.

## Validation Procedure

1. A collection of unseen moon images encompassing a diverse range of moon phases was compiled.
2. The images were fed into the trained moon phase classification model.
3. The model's predicted moon phase for each image was recorded.
4. The predicted phases were then manually compared to the ground truth (actual moon phase) of each image by a human expert.

For the majority of the unseen images, the model's predictions aligned with the human expert's assessment.

```
classify_images('tested images/fm.jpeg')

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step

'The Image belongs to full moon with a score of 91.07456803321838'
```

```
classify_images('tested images/wxc.jpeg')

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step

'The Image belongs to waxing crescent with a score of 93.75767707824707'
```

```
classify_images('tested images/waning-gibbous.jpg')

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step

'The Image belongs to waning gibbous with a score of 97.28456735610962'
```

```
classify_images('tested images/third-quarter.jpg')

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step

'The Image belongs to third quarter with a score of 72.85491824150085'
```

```
classify_images('tested images/fr.jpeg')

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step

'The Image belongs to first quarter with a score of 96.91020250320435'
```

**Fig 13: The result of manually validated images**

### 4.2 Comparison with Existing Techniques

The exploration of moon phase classification using deep learning presented in this report is a valuable contribution to the field of lunar image analysis. However, it's crucial to situate this work within the broader context of existing techniques. Here, we compare our CNN-based approach with some established methods:

**Traditional Image Processing Techniques**:

- **Thresholding**: This technique utilizes a threshold value to segment the moon from the background. While simple and efficient, it can struggle with images containing noise or variations in lighting conditions.
- **Edge Detection**: Algorithms like Canny edge detection can be employed to identify the moon's prominent edges. However, this approach might not be effective for all moon phases, particularly those with subtle edge variations.
- **Template Matching**: Predefined templates representing different moon phases can be used for comparison with the target image. However, this method requires a large set of accurate templates and may not account for natural variations within each phase.

**Machine Learning Techniques**:

- **Support Vector Machines (SVMs)**: SVMs can be trained to classify moon phases based on extracted image features. However, they often require careful feature engineering and might not scale well with a large number of moon phase classes.
- **Random Forests**: These ensemble methods combine multiple decision trees for classification. While robust to overfitting, they can be less interpretable compared to CNNs, making it challenging to understand the rationale behind the model's predictions.

**Deep Learning Techniques**:

The work leverages a Convolutional Neural Network (CNN) for moon phase classification. Compared to the aforementioned techniques, CNNs offer several advantages:

- **Automatic Feature Extraction**: CNNs automatically learn relevant features from the moon images during the training process, eliminating the need for manual feature engineering.
- **Improved Generalizability**: By learning from a diverse dataset, CNNs can effectively adapt to variations in lighting, noise, and camera angles, leading to improved performance on unseen data.
- **State-of-the-Art Performance**: Deep learning techniques have consistently achieved high accuracy in various image classification tasks, including moon phase detection. This report contributes to this growing body of research.

**Limitations of the Current Study**:

It's important to acknowledge the limitations of the current study:

- **Dataset Size**: The performance of a deep learning model can be significantly influenced by the size and diversity of the training dataset. While our dataset encompassed a range of moon phases, it might be beneficial to explore even larger datasets in future work.
- **Model Complexity**: The complexity of the CNN architecture employed in this work can be further optimized. Exploring different network architectures and hyperparameter tuning techniques could potentially lead to improved accuracy.

# CHAPTER 5: CONCLUSION AND FUTURE SCOPE

## 5.1 CONCLUSION

This report has delved into the exploration of a Convolutional Neural Network (CNN) for the task of moon phase classification. We meticulously designed a methodology encompassing data acquisition, preprocessing, model creation, training, and evaluation. The heart of the system lies in the CNN architecture, which is adept at extracting features from lunar images and effectively classifying them into distinct moon phases. The model was rigorously evaluated using various metrics, achieving promising results with an overall accuracy of 73%. The classification report provided a detailed breakdown of the model's performance for each individual moon phase class, highlighting strengths and weaknesses. Analysis of the accuracy and loss curves revealed the model's ability to learn and generalize effectively. The confusion matrix offered valuable insights into potential misclassifications, pinpointing areas for further improvement. Finally, manual validation with unseen images served as a final check on the model's generalizability to real-world scenarios.

## 5.2 FUTURE SCOPE

Building upon the foundation established in this project, several promising avenues for future exploration exist:

- **Transfer Learning**: Utilizing pre-trained CNN models on large image datasets like ImageNet as a starting point for moon phase classification could potentially improve performance with a smaller, specialized moon image dataset.
- **Multi-task Learning**: The model could be extended to not only classify moon phases but also estimate additional lunar properties like illumination percentage or atmospheric effects.
- **Real-Time Applications**: Integrating the moon phase classification model with mobile applications or astronomical software could empower users with real-time moon phase information.

By continuously refining the model and exploring these exciting advancements, we can further enhance the accuracy and robustness of deep learning-based moon phase detection while expanding its applications in various fields.

**REFERENCES**


[1] Xinyu Lei, Hongguang Pan, & Xiangdong Huang. A Dilated CNN Model for Image Classification. *IEEE*

[2] Ata Jahangir Moshayedi, Zu-yan Chen, Liefa Liao, & Shuai Li. Sunfa Ata Zuyan machine learning models for moon phase detection: algorithm, prototype and performance comparison. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*

[3] Saroj K. Mehra, & Ganapati Panda. Deep learning in astronomy: a tutorial perspective. *Published in 2021 Volume 230, pages 2285 – 2317, The European Journal Special Topics*

[4] Ajit Kembhavi, & Rohan Pattnaik. Machine learning in astronomy. *Published in 2022, Volume 43, article number 76, Journal of Astrophysics and Astronomy*