Recognising Mythical Creatures Using Convolutional Neural Networks.

Nitish Kumar Jha

MSc in Computer Science
The University of Bath
2023-2024

Recognising Mythical Creatures Using Convolutional Neural Networks.

Submitted by: Nitish Kumar Jha

Copyright

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Master of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

This dissertation presents a novel approach to classifying mythological creatures using Convolutional Neural Networks (CNNs), specifically MobileNetV2, enhanced through transfer learning. The challenges in this classification task stem from the limited availability of annotated datasets and the significant variability in artistic representations of these creatures across cultures and time periods. The study tackles these challenges by fine-tuning a pre-trained MobileNetV2 model to classify five mythological creatures—Centaur, Griffin, Hydra, Minotaur, and Pegasus—using a custom dataset of 100 images.

Transfer learning, combined with data augmentation techniques such as rotation and color jittering, was employed to address the small dataset size and improve model generalisation. The final model achieved a classification accuracy of 70% and demonstrated balanced performance across classes, with a macro F1 score of 70.29%. Despite these promising results, the study also highlights the limitations posed by the small dataset and the difficulties in classifying creatures with similar visual features.

These findings suggest that CNNs, particularly MobileNetV2, are viable tools for classifying culturally significant imagery. The implications of this research extend to fields such as digital humanities and cultural heritage preservation, where machine learning can aid in the documentation and categorisation of mythological and historical artifacts.

Contents

1 Acknowledgements **vii**

2 Introduction **1**
2.1 Background . . . 1
2.2 Problem Statement . . . 2
2.3 Research Objectives . . . 3

3 Literature Review **4**
3.1 Overview of Image Classification . . . 4
3.2 Convolutional Neural Networks in Image Classification . . . 4
3.3 MobileNetV2 Architecture . . . 5
3.3.1 Inverted Residuals and Linear Bottlenecks . . . 5
3.3.2 Depthwise Separable Convolutions . . . 6
3.3.3 Overall Architecture . . . 6
3.3.4 Performance and Applications . . . 6
3.4 Applications of CNNs in Image Classification . . . 7
3.4.1 Object Detection and Recognition . . . 7
3.4.2 Facial Recognition . . . 8
3.4.3 Arts and Cultural Heritage . . . 9
3.4.4 Conclusion . . . 10
3.5 Mythological Creatures in Image Classification . . . 10
3.5.1 Challenges in Classifying Mythological Creatures . . . 10
3.5.2 Transfer Learning and Data Augmentation . . . 11
3.5.3 CNN Architectures for Mythological Creature Classification . . . 11
3.5.4 Recent Approaches in Mythological Creature Classification . . . 12
3.5.5 Conclusion . . . 12
3.6 Summary of Literature Review . . . 12

4 Methodology **14**
4.1 Dataset Collection . . . 14
4.1.1 Data Sources . . . 14
4.1.2 Dataset Composition . . . 14
4.1.3 Image Annotation for YOLOv8 . . . 15
4.1.4 Dataset Structuring for CNN . . . 15
4.1.5 Challenges Limitations . . . 16
4.2 Model Selection . . . 16
4.2.1 Initial Model Selection: YOLOv8 for Object Detection . . . 16

4.2.2 Final Model Selection: MobileNetV2 for Image Classification 16
4.2.3 Justification for Model Transition 17
4.3 Model Architecture 17
4.3.1 Initial Model Architecture and Hyperparameter Tuning 17
4.3.2 Issues with Overfitting 18
4.3.3 Customisation for Mythological Creature Classification 19
4.3.4 Model Implementation 19
4.3.5 Training and Regularisation 20
4.4 Training the Model 21
4.4.1 Training Parameters 21
4.4.2 Data Augmentation 21
4.4.3 Training Procedure 22
4.5 Evaluation Metrics 23
4.5.1 Accuracy 23
4.5.2 Precision 24
4.5.3 Recall 24
4.5.4 F1 Score 24
4.5.5 Confusion Matrix 24
4.5.6 Classification Report 25

5 Results and Discussion **27**
5.1 Model Performance 27
5.1.1 Training and Validation Accuracy 27
5.1.2 Training and Validation Loss 27
5.1.3 Confusion Matrix 28
5.1.4 Evaluation Metrics 28
5.1.5 Overfitting and Generalisation 29
5.2 Impact of Data Augmentation 29
5.2.1 Types of Data Augmentation Applied 30
5.2.2 Reduced Overfitting 30
5.2.3 Considerations for Future Work 30
5.3 Discussion on Results 30
5.3.1 Strengths of the Final Model 30
5.3.2 Strengths of the Complex Model 30
5.3.3 Limitations of the Models 31
5.3.4 Areas for Improvement 31
5.3.5 Conclusion on Model Selection 31

6 Conclusions **32**
6.1 Summary of Findings 32
6.2 Contributions to the Field 33
6.3 Limitations of the Study 33
6.4 Future Work 34

Bibliography **36**

A Appendix: Evaluation Metrics **41**
A.1 Accuracy 41
A.2 Precision 41

A.3 Recall (Sensitivity or True Positive Rate) 42
A.4 F1 Score 42
A.5 Overall Score 42

B Appendix: Image Generation and Visualisation 44
B.1 Image Generation and Visualisation 44
B.1.1 Loading and Preprocessing the Image 44
B.1.2 Visualising the Original Image and Its RGB Channels 44
B.1.3 Visualising Activations of the First Convolutional Layer 44
B.1.4 Figures 45

C User Documentation 47

D Raw Results Output 51
D.1 GitHub Repository Link 51
D.2 Validation Result 51
D.3 Prediction Result 51

List of Figures

3.1 Inverted Residual Block with Linear Bottleneck in MobileNetV2 (Sandler et al., 2018) . . . 5
3.2 Depthwise Separable Convolution (Pandey, 2018) . . . 6
3.3 MobileNetV2 Architecture Overview (Sandler et al., 2018) . . . 7
3.4 Object Detection using Faster R-CNN (Source: Girshick et al. (2014)) . . . 8
3.5 Facial Recognition Pipeline using CNNs (Source: (Taigman et al., 2014)) . . . 9
3.6 Style Transfer using CNNs (Source: Gatys, Ecker and Bethge (2016)) . . . 10
3.7 Data Augmentation Example for Mythological Creature Classification Singh (2022) . . . 11
3.8 CNN Feature Maps for Mythological Creatures Karpathy, Johnson and Li (2024) 12

4.1 Distribution of images across classes . . . 15
4.2 Summary of the customised MobileNetV2 model used for mythological creature classification . . . 17
4.3 An illustration of how to apply data augmentation techniques to images. . . . 22
4.4 The accuracy and loss curves for training and validation over time. . . . 23
4.5 CNN Confusion Metrix . . . 25

5.1 Loss and Accuracy curve for both the models . . . 28
5.2 Confusion Matrix . . . 29

B.1 Viewing the original image and its RGB channels. While the top-left image displays the original image, the other three images show separate channels for red, green, and blue. . . . 46
B.2 The activations from the first convolutional layer can be visualised. Each sub-image is a representation of a different filter in the first layer, which highlights various features that the model detects. . . . 46

D.1 Image validation prediction output. . . . 52
D.2 Image test prediction output. . . . 53

List of Tables

4.1 Hyperparameters and their ranges for initial model tuning Nazir, Patel and Patel (2018). 18
4.2 Training parameters used for the final CNN model. 21
4.3 Overall performance metrics for the CNN model. 25
4.4 Classification report for the mythological creatures dataset. 26

5.1 Summary of Evaluation Metrics for Final and Complex Models 29

Chapter 1

Acknowledgements

The successful completion of this project was much dependent on the invaluable assistance provided by numerous individuals. Gratitude is extended to my supervisor, Prof. Peter Hall, for his guidance during this study, as well as to Google Colab for providing the complimentary GPU.

Chapter 2

Introduction

2.1 Background

Although robots have made significant advancements in their capacity to identify and categorise photographs, particularly in classifying photos related to common items or animals, the performance of these technologies on more complex issues remains to be clearly demonstrated. Therefore, the task at hand is to apply sophisticated machine learning methods to cultural mythology in order to develop a reliable image classifier that can accurately identify mythical creatures. Mythology, having been ingrained in human society for millennia, encompasses a great array of creatures like Centaurs, Griffins, Hydras, Minotaurs, and Pegasus. These symbols have been employed in ancient mythology, as well as shown in illustrations and literary works. Given the distinctiveness possessed by each mythological monster, they are very suitable for image categorisation objectives. Given the limited availability of comprehensive and annotated datasets for mythological animals, it becomes even more intricate and difficult to develop a precise classification model. This necessitates innovative approaches to use machine learning techniques Tan and Le (2019).

The application of deep learning techniques, specifically CNNs, has revolutionised approaches to picture classification. CNNs exhibit greater efficacy in autonomously acquiring features on a spatial hierarchy as compared to conventional machine learning approaches in several image identification applications. Furthermore, the effectiveness of this approach has been significantly enhanced with the introduction of transfer learning, which addresses the limited availability of labelled data. Consequently, this method serves as a means of refining pre-trained models for novel tasks, thereby creating intriguing opportunities for the implementation of advanced picture classification algorithms in certain fields, such as the recognition of legendary animals.

The present work leverages the advantages of CNNs and transfer learning to construct a comprehensive picture classifier specifically designed for mythological creatures. We train our model using a bespoke dataset consisting of photos that represents five distinct mythological figures. The selected CNN is a modified iteration of MobileNetV2, a compact deep neural network that has shown outstanding performance with comparatively less computational demands in comparison to other models from its era Howard et al. (2017).

The image classifier in our system will be constructed, trained, and deployed using TensorFlow Hub. It is a collection of universally applicable machine learning modules, which greatly

facilitate the integration of pre-trained models. Another crucial component of this job is the evaluation and interpretation of the model, which involves visualising the predictions and analysing the corresponding confidence level.

The major objectives of this experiment are two-fold: to achieve high accuracy in classifying photos of mythological creatures and, equally important, to provide a practical proof of the usefulness of transfer learning for specialised tasks with little data. Sustained establishment of the model would have significant ramifications for analogous applications in digital humanities and cultural heritage conservation, since machine learning would now be capable of categorising and safeguarding historical artefacts and works of art.

The intriguing convergence of cutting-edge technology with the timeless fabric of cultural tradition transcends being merely a technical obstacle. Instructing the machine to identify those beings conceived from human imagination not only expands the boundaries of artificial intelligence but also establishes a connection between our mythical history and our technological future.

An analysis of the most recent advancements in employing CNN and Transfer Learning for accurate picture categorisation has been conducted in Section 2. Section 3 of our paper provides a detailed description of our approach, including dataset preparation, model design, and training techniques. Section 4 presents the outcomes of our experiments together with their analysis. In conclusion, Section 5 examines the consequences of our discoveries, acknowledges the constraints, and proposes avenues for further investigation in this captivating field where technology intersects with mythology.

2.2 Problem Statement

The categorisation of mythological subjects in visual representations poses an intricate and diverse problem in the field of computer vision and machine learning. The present study aims to tackle the following focal points:

1. **Data insufficiency and complexity:** The limited availability of extensive, annotated datasets particularly dedicated to mythological animals greatly impedes the progress of constructing reliable classification models. In contrast to well-established categories in conventional image classification experiments, mythological beings are found in a wide range of artistic styles, cultural contexts, and historical eras, resulting in significant variation within each class and similarities between different classes.
2. **Domain-Specific Transfer Learning:** Although transfer learning has demonstrated potential in several image classification problems, its implementation in the specialised field of mythological creatures necessitates meticulous adjustment. Novel techniques are required to fine-tune pre-trained models in order to balance the preservation of general traits with the acquisition of domain-specific information, especially considering the distinctive qualities of mythical imagery.
3. **Representation Learning Challenges:** Extraction of features and learning of representations are greatly challenged by the beautiful and frequently abstract character of mythological creature depictions. The acquisition of tiny, style-invariant elements that differentiate various mythological beings may provide a challenge for conventional CNN designs.

4. **Confidence Calibration:** High-confidence misclassifications pose significant challenges in this field, since the demarcation between classes might be subjective and influenced by cultural biases. Accurate calibration of model confidence and effective management of uncertainty are essential for ensuring the dependability and comprehensibility of the categorisation system.
5. **Cross-Cultural Generalisation:** Another layer of intricacy arises from the diverse cultural origins of mythological animals. Models trained on representations from a single cultural context may not effectively generalise to depictions from other cultures, thereby requiring methods that can capture culturally invariant characteristics.

2.3 Research Objectives

The proposed study aims to address the issues mentioned above by pursuing the following objectives:

1. **Dataset Development**:
   - Collect a diverse set of mythological creature images.
   - Use advanced data augmentation techniques to improve model robustness.
   - Standardise annotation across classifications.
2. **Architecture and Transfer Learning**:
   - Compare CNN architectures (e.g., MobileNetV2, EfficientNet, Vision Transformer).
   - Propose transfer learning techniques for mythological imagery.
   - Explore multi-task learning for related categories.
3. **Feature Representation**:
   - Visualise learned features and improve model interpretability using attention mechanisms or explainable AI techniques.
4. **Confidence Calibration**:
   - Implement uncertainty quantification techniques to improve model reliability.
   - Develop a framework to align model confidence with performance.
5. **Cultural Generalisation**:
   - Assess model performance across diverse cultural depictions.
   - Develop methodologies for domain adaptation and a standardised assessment approach considering cultural differences.

This project intends to enhance the field of picture categorisation in culturally diverse and data-limited areas, while also augmenting the larger knowledge of transfer learning, uncertainty quantification, and cross-cultural AI applications.

Chapter 3

Literature Review

3.1 Overview of Image Classification

Image classification in computer vision is the process of classifying images based on their content. Deep learning approaches have superseded SVMs and k-Nearest Neighbors (k-NN) for this problem Lowe (1999). Historical techniques were based on handcrafted traits, which made them rather difficult to generalise.

CNNs changed the field when they became available since they could learn hierarchical visual features on their own (Lecun et al., 1998). Notable architectures like as AlexNet, VGGNet, and ResNet have progressively increased the classification tasks' accuracy.

Models like MobileNet Howard et al. (2017) and EfficientNet (Tan and Le, 2019), which seek to strike a compromise between high accuracy and computational efficiency, are at the center of recent advances in network efficiency optimisation. Performance is greatly enhanced by transfer learning and attention techniques like Vision Transformers (ViTs) (Dosovitskiy et al., 2020), particularly when used on smaller datasets with specific application in mind.

Not withstanding these progresses, there are still constraints in precise classification and limited availability of data. The current review critically examines advances in image categorisation, evaluates the current applications, and underlines the continuing obstacles in this area.

3.2 Convolutional Neural Networks in Image Classification

CNNs have revolutionised image classification by using a direct learning method to hierarchical features extracted from raw image data, therefore removing the need for manually designed features. The visual brain serves as the inspiration for CNNs, which include convolutional layers to identify local patterns, pooling layers to decrease dimensionality, and fully connected layers to generate final predictions. Refer to appendix B.1 for more insight on how CNN classifies images into RGB channels.

Pioneering CNNs like LeNet-5 Lecun et al. (1998) established the basis for more advanced models. In the ImageNet competition, AlexNet Krizhevsky, Sutskever and Hinton (2012) attained a top-5 error rate of 15.3% by using innovative techniques such as ReLU activations

and dropout, significantly improving the precision of image categorisation. The VGGNet model Simonyan and Zisserman (2014) demonstrated the benefits of elaborate networks consisting of 16 to 19 layers, although at a higher computational cost. In order to tackle the problem of vanishing gradients, the ResNet model integrated residual connections, enabling the construction of deeper networks and yielding a top-5 error rate of 3.6%.

Newer architectures like DenseNet and Inception have improved the performance of CNNs by implementing more efficient information transfer and multi-scale feature capture. transfer learning has enhanced the efficacy of CNNs by modifying pre-trained models, such as those from ImageNet, to suit new tasks, therefore decreasing training time and enhancing outcomes on smaller datasets. These developments validate CNNs as the dominant method in picture categorisation.

3.3 MobileNetV2 Architecture

MobileNetV2 is a type of convolutional neural network (CNN) optimised for mobile and embedded applications Sandler et al. (2018). It builds on MobileNetV1 by balancing model accuracy with computing efficiency. A key feature is the inverted residual block with a linear bottleneck, which reduces parameter count and computational cost while maintaining accuracy.

3.3.1 Inverted Residuals and Linear Bottlenecks

The inverted residual block reduces the input dimensionality with a 1x1 convolution ("bottleneck" layer), applies depthwise separable convolution, and expands the output with another 1x1 convolution, forming a residual connection. This structure allows for efficient feature extraction with fewer parameters.

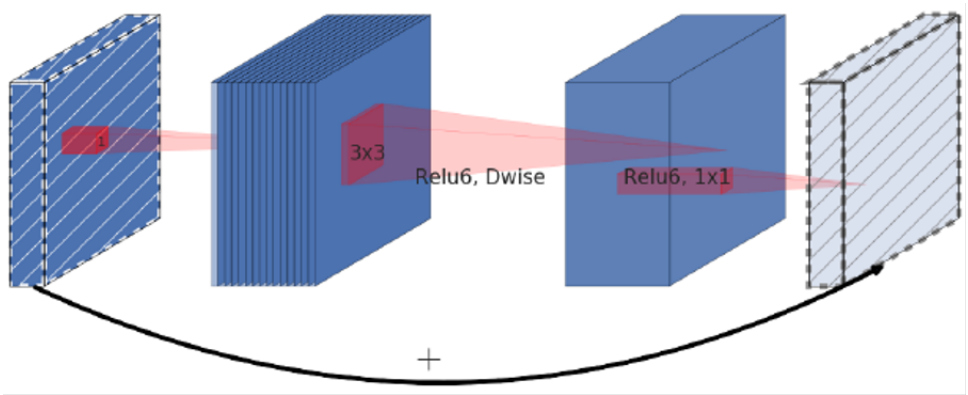


Figure 3.1: Inverted Residual Block with Linear Bottleneck in MobileNetV2 (Sandler et al., 2018)

The sequence can be expressed as:

$$Output = \mathbf{X} + f\left(\sigma\left(\mathbf{W}_2 \cdot (\mathbf{W}_1 \cdot \mathbf{X})\right)\right)$$

where $\mathbf{X}$ is the input feature map, $f(\cdot)$ is the depthwise convolution, and $\mathbf{W}_1$, $\mathbf{W}_2$ are weight matrices.

3.3.2 Depthwise Separable Convolutions

A key feature of MobileNetV2 is depthwise separable convolution, which divides a normal convolution into two operations: depthwise convolution (one filter per input channel) and pointwise convolution (1x1 to gather the outputs).This reduces the computational cost compared to standard convolution:

$$Cost standard = H \times W \times k \times k \times Din \times D_{out}$$

$$Cost depthwise = H \times W \times (k \times k \times Din + D_{in} \times D_{out})$$

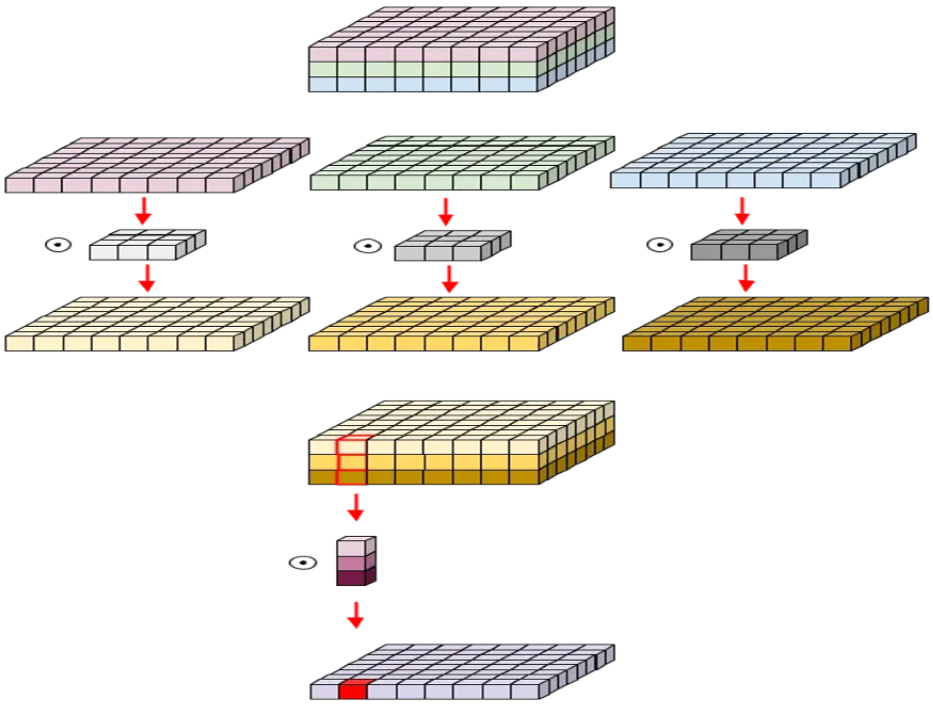


Figure 3.2: Depthwise Separable Convolution (Pandey, 2018)

3.3.3 Overall Architecture

Global average pooling and a fully connected layer come last in MobileNetV2's architecture, which starts with a fully convolutional layer and continues with inverted residual blocks. It is appropriate for a range of applications, including cloud-based systems and mobile devices, thanks to its scalable design, which enables customisation in depth, width, and resolution.

3.3.4 Performance and Applications

Compared to models such as VGG16 and ResNet, MobileNetV2 achieves 72% accuracy on ImageNet using less parameters. Its applications span from object identification to transfer learning in areas like autonomous vehicles and medical imaging. Its balance of efficiency and accuracy makes it ideal for resource-constrained environments Sandler et al. (2018).

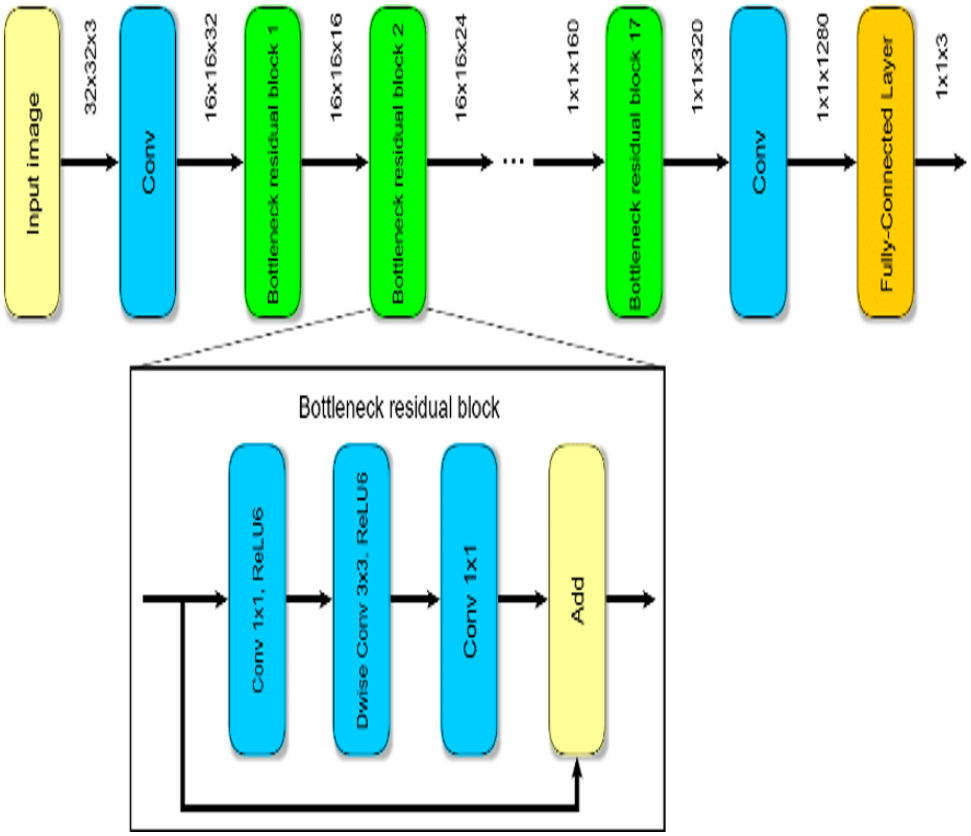


Figure 3.3: MobileNetV2 Architecture Overview (Sandler et al., 2018)

3.4 Applications of CNNs in Image Classification

Recent image categorisation using CNNs has made major progress possible in many different fields. Although CNNs have advanced object recognition, medical imaging, and autonomous automobiles, this paper will mostly address its application in object detection and recognition, facial recognition, and the visual arts and cultural legacy. CNNs have shown incredible ability to extract complex patterns from visual data, therefore increasing both the theoretical and practical aspects of computer vision.

3.4.1 Object Detection and Recognition

CNNs find extensive use in the field of object detection, which is the recognition and exact localisation of objects inside images. First presented by Girshick et al. (2014), the Region-based CNN is a fundamental method in this field. By combining area suggestions with CNNs, the R-CNN algorithm precisely classifies objects within the suggested regions, hence significantly improving detection accuracy.

In CNNs, object detection is based on the mathematical principle of reducing a multi-task loss that integrates both classification and localisation tasks:

$$L(p, t) = L_{cls}(p, p^{)} + \lambda \cdot L_{reg}(t, t^{)}$$

Where:

- $L_{cls}(p, p^{)}$ is the classification loss (e.g., cross-entropy) between the predicted class scores p and the true labels p.
- $L_{reg}(t, t^{)}$ is the regression loss (e.g., smooth L1 loss) between the predicted bounding box coordinates t and the ground truth coordinates t.
- λ is a balancing parameter.

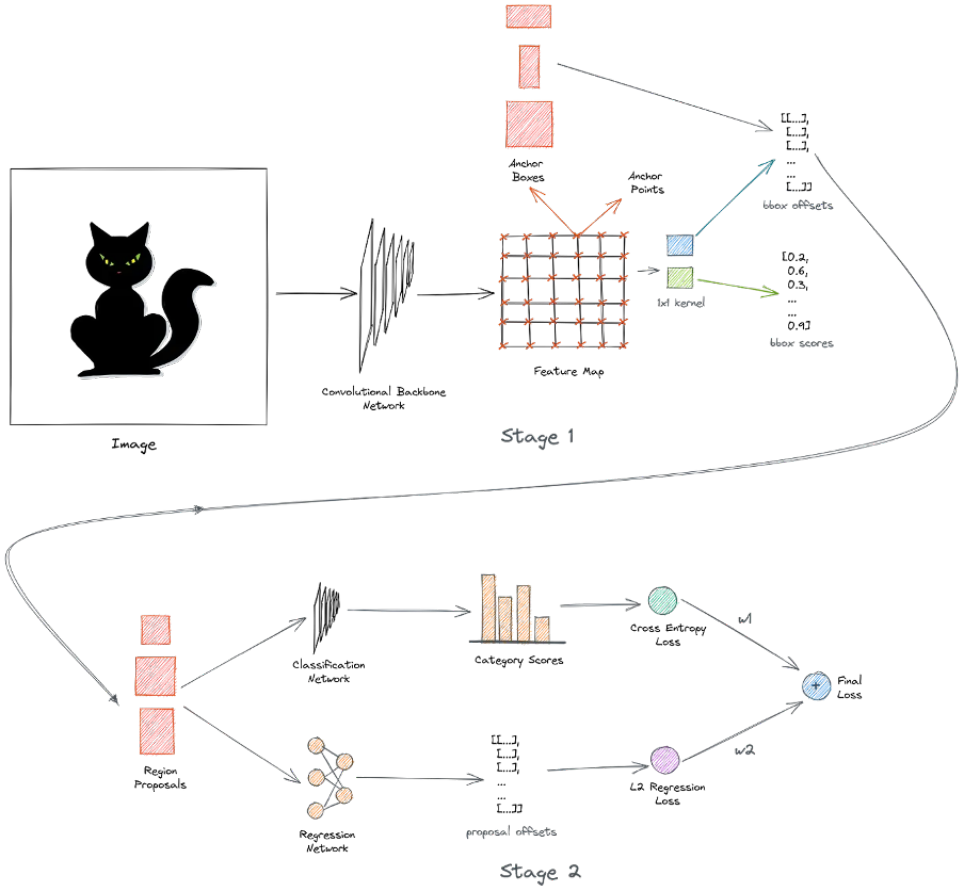


Figure 3.4: Object Detection using Faster R-CNN (Source: Girshick et al. (2014))

Object detection models like YOLO (You Only Look Once) Redmon et al. (2015) and SSD (Single Shot MultiBox Detector) Liu et al. (2015) are meticulously designed for real-time applications. Various applications, such as autonomous driving, surveillance, and robotics, are ideally suited to the previously described models, which establish an equilibrium between speed and precision.

3.4.2 Facial Recognition

Especially in the field of facial recognition, CNNs find great application in safety systems, user identification, and social media platforms. CNNs extract and compare facial traits, therefore converting images into a feature space in which like faces are aggregated. Metric learning approaches, including triplet loss, are frequently used to improve this process by ensuring that the model can differentiate between distinct individuals.

The loss function for facial recognition typically takes the form:

$$L = \max(0, m + D(f(\mathbf{x}_a), f(\mathbf{x}_p)) - D(f(\mathbf{x}_a), f(\mathbf{x}_n)))$$

Where:

- $f(\mathbf{x})$ is the feature representation of image $\mathbf{x}$ learned by the CNN.
- $D(\cdot, \cdot)$ is a distance metric (e.g., Euclidean distance).
- $\mathbf{x}_a$, $\mathbf{x}_p$, and $\mathbf{x}_n$ are the anchor, positive, and negative examples, respectively.
- m is a margin parameter ensuring that dissimilar pairs are adequately separated in the feature space.

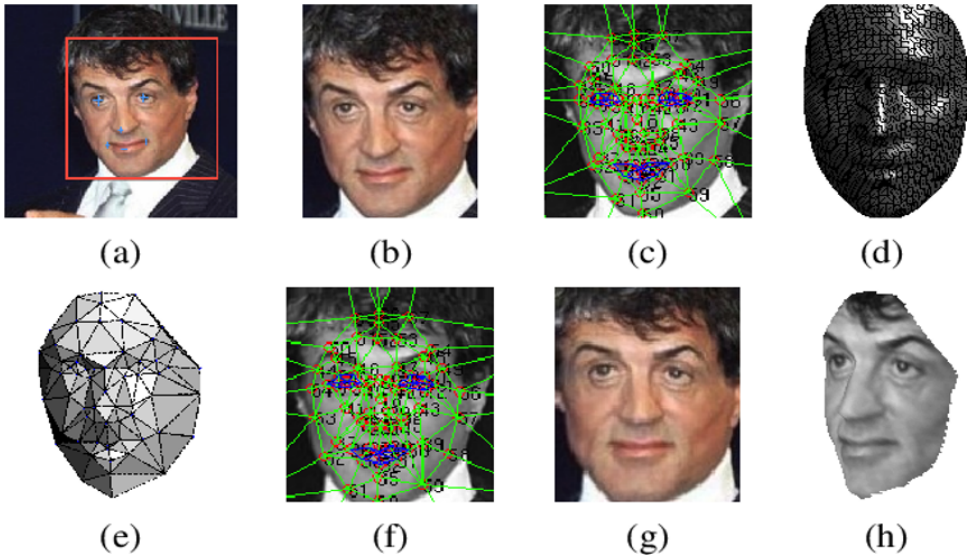


Figure 3.5: Facial Recognition Pipeline using CNNs (Source: (Taigman et al., 2014))

CNNs employed for facial recognition are used in various applications, such as smartphone unlocking and individual monitoring in surveillance footage. This elicits considerable apprehensions pertaining to privacy and ethical usage.

3.4.3 Arts and Cultural Heritage

In addition to traditional areas, deep neural networks have found applications in many industries like the arts and cultural heritage sectors. CNNs are also extensively used in applications like artwork categorisation, style transfer, and image restoration. In CNNs, style transfer denotes the integration of the stylistic attributes of one image with the content of another image. An ideal loss function is employed to equilibrate the representations of both content and style, derived from various levels of the network Gatys, Ecker and Bethge (2016).

$$L_{total} = \alpha L_{content} + \beta L_{style}$$

Where:

- $L_{content}$ measures the difference between the content of the input and generated images.

- L_{style} measures the difference in style between the input and the reference style image.
- α and β are weighting factors.

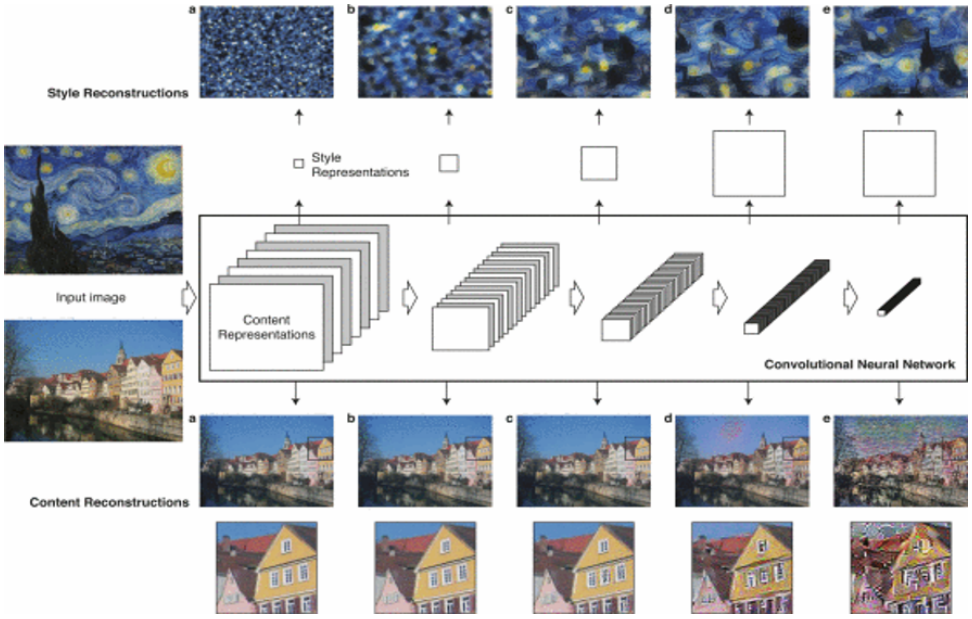


Figure 3.6: Style Transfer using CNNs (Source: Gatys, Ecker and Bethge (2016))

Furthermore, CNNs are used for digitising and categorising artworks, consequently enhancing collection management efficiency for museums and galleries and increasing public access to cultural material. These examples show how CNNs can be used to solve complex visual challenges that go beyond conventional image matching.

3.4.4 Conclusion

Emerging as a vital tool in image categorisation, CNNs help to advance several fields. In disciplines including real-time object detection, facial recognition, and the arts and cultural legacy, CNNs offer remarkable accuracy and efficiency. Their application in several different fields emphasises their ability to affect many aspects of human life, therefore extending the possibilities for machine learning to impact many spheres of existence.

3.5 Mythological Creatures in Image Classification

Classifying mythological creatures presents challenges due to the varied representations across cultures and artistic styles. Unlike conventional objects, mythological creatures lack uniform visual characteristics, making classification complex. The limited availability of annotated datasets further highlights the need for transfer learning and data augmentation. Characters like Pegasus, Griffin, and Hydra often have abstract and inconsistent depictions, complicating machine learning efforts.

3.5.1 Challenges in Classifying Mythological Creatures

1. **Intra-Class Variability:** Mythological creatures, such as the Griffin, vary significantly across cultures and time periods. For example, the Minotaur is depicted with a human

body and bull's face in ancient Greek art but appears more anthropomorphised in Renaissance depictions.

2. **Inter-Class Similarity:** Mythological creatures like the Griffin and Sphinx share many visual characteristics, making differentiation challenging for CNNs.
3. **Lack of Labeled Datasets:** Unlike common objects, mythological creatures are not well-represented in large, annotated datasets like ImageNet. Transfer learning from large datasets can address this limitation.

3.5.2 Transfer Learning and Data Augmentation

Transfer learning refines pre-trained models on large datasets, such as ImageNet, to fit domain-specific tasks. This technique enables CNNs to recognise broader patterns and apply them to mythological creatures. The fine-tuning process is mathematically expressed as:

$$L(\theta) = L_{task}(\theta_{new}) + \lambda L_{pretrain}(\theta_{old})$$

Where θ_{new} represents the parameters fine-tuned on the new dataset, and λ balances new learning with pre-trained knowledge.

Data augmentation further improves the model's generalisation by introducing transformations such as rotation, color jitter, and random cropping. These techniques help adapt the model to the diverse artistic styles of mythological creatures.

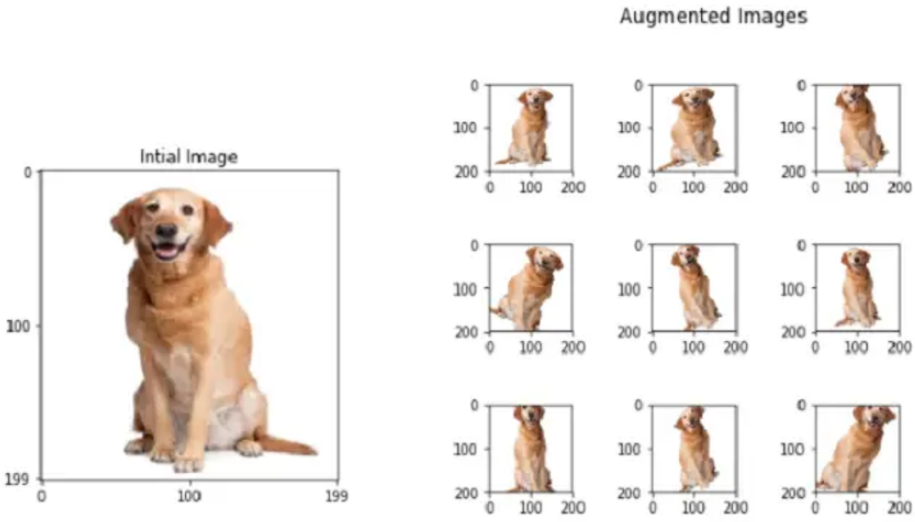


Figure 3.7: Data Augmentation Example for Mythological Creature Classification Singh (2022)

3.5.3 CNN Architectures for Mythological Creature Classification

CNNs, like MobileNetV2 and EfficientNet, are effective for identifying mythical entities owing to their capacity to learn hierarchical characteristics. MobileNetV2 employs depthwise separable convolutions and inverted residuals, rendering it computationally efficient for small datasets. The feature extraction process in CNNs is represented as:

$$f(\mathbf{x}) = W * \mathbf{x} + b$$

Where $f(\mathbf{x})$ is the feature map learned from the input image $\mathbf{x}$, W is the filter applied, and b is the bias.

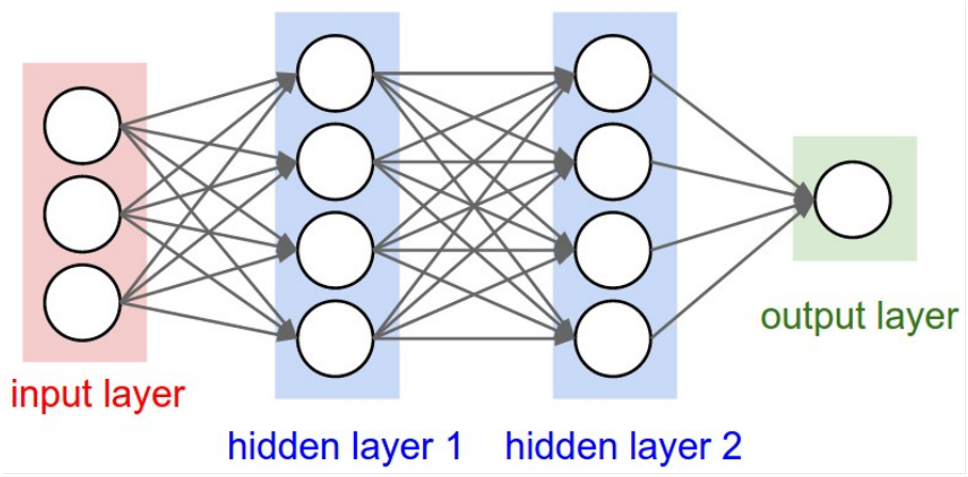


Figure 3.8: CNN Feature Maps for Mythological Creatures Karpathy, Johnson and Li (2024)

3.5.4 Recent Approaches in Mythological Creature Classification

Recent approaches have explored multimodal techniques, integrating visual data with textual descriptions or metadata to improve classification accuracy. Generative models like GANs have also been used to create artificial mythological images, helping balance datasets where certain creatures are underrepresented.

3.5.5 Conclusion

Classifying mythological creatures with CNNs is complex due to their diverse representations and limited datasets. Using techniques like transfer learning and data augmentation, along with efficient architectures like MobileNetV2, can enhance classification accuracy and contribute to preserving cultural heritage through technology.

3.6 Summary of Literature Review

The literature review conducted a comprehensive analysis of the development and utilisation of CNNs in the categorisation of images, emphasising its revolutionary influence in several domains.

Historically, image classification approaches mainly depended on manually designed features, such as edge detection and texture analysis, which had restricted capacity to generalise across various datasets Lowe (1999). The advent of CNNs transformed the discipline by mechanically extracting features and facilitating the acquisition of hierarchical representations from data. This greatly enhanced the precision and resilience of image classification systems Krizhevsky, Sutskever and Hinton (2012), Lecun et al. (1998).

Numerous critical CNN architectures have significantly influenced the domain of picture categorisation. LeNet, an early convolutional neural network (CNN) model, established the foundation for further advancements Lecun et al. (1998). A significant milestone was reached in 2012 when AlexNet achieved success in the ImageNet competition, demonstrating the immense capabilities of deep learning Krizhevsky, Sutskever and Hinton (2012). Later architectures like as VGGNet and ResNet implemented more complex networks and incorporated novel features like residual connections, therefore expanding the limits of performance Simonyan and Zisserman (2014), He et al. (2016). The efficiency-oriented MobileNetV2 model showcased the possibility of attaining excellent accuracy even on devices with limited resources Sandler et al. (2018).

CNNs have shown their effectiveness in various domains such as object identification, medical imaging, autonomous cars, and facial recognition. They have smoothed the way for the automatic identification and precise localisation of objects in images Girshick et al. (2014). CNNs have been used in medical imaging to accurately categorise diseases, similar to that of human specialists Esteva et al. (2017). CNNs are essential for autonomous cars to perform important functions such as object recognition and scene interpretation, which are crucial for ensuring safe navigation Bojarski et al. (2016). Facial recognition and style transfer are two other domains in which CNNs have shown their immense value, making significant contributions to the fields of security and digital art Taigman et al. (2014), Gatys, Ecker and Bethge (2016).

The categorisation of mythological beings poses distinct difficulties because of their widespread and culturally disparate depictions. The presence of significant intra-class variability and inter-class similarities poses challenges for conventional models in distinguishing between species such as the Griffin and Sphinx. Furthermore, the limited availability of annotated datasets adds complexity to the process of training the model. methodologies such as transfer learning and data augmentation have played a crucial role in tackling these difficulties, enabling models to be refined on little datasets while preserving a high level of accuracy Yosinski et al. (2014).

While much progress has been achieved, obstacles persist, especially in specialised domains such as the categorisation of mythological creatures. It is emphasised in the literature that larger and more varied datasets, as well as enhanced data augmentation methods, are necessary to increase model performance. Furthermore, the investigation of multimodal methodologies and the incorporation of generative models may provide novel opportunities for study, and perhaps surmount existing constraints Tan and Le (2019).

Enhancing the efficiency of CNN architectures, especially for implementation in contexts with limited resources, is a crucial focal point for future study. Neural architecture search (NAS) and model quantisation provide intriguing avenues for developing more efficient, task-specific models Sandler et al. (2018).

Ultimately, the literature literature review emphasises the substantial influence of CNNs on the task of image classification and brings attention to the persistent difficulties in specific domains. Sustained advancement in transfer learning, data augmentation, and multimodal techniques will be essential for the progress of the profession.

Chapter 4

Methodology

4.1 Dataset Collection

This study's foundation is a large dataset that was hand-picked to make it easier to classify mythological monsters using CNNs and YOLOv8 for object detection. Given the unique qualities of the target classes—mythological creatures—special consideration was given to determining the dataset's relevance, diversity, and quality.

4.1.1 Data Sources

The images used in this study were taken from publicly accessible websites, with a focus on two primary sites: Google Photos and Mythopedia.com. These sources were chosen because they provide a wide range of unique visual renderings of mythological creatures, which are crucial for building an extensive dataset.

- **Google Images:** With its extensive image search, Google provides a variety of depictions for every species, from classic drawings to modern artistic interpretations. We gathered a wide variety of images by using specific keywords for each creature (e.g., "Centaur," "Pegasus").
- **Mythopedia.com:** Offering an extensive selection of mythology-related content, Mythopedia.com offers superbly curated, conceptually cohesive images. Using this site to gather images of mythological animals that are accurate and culturally significant turned out to be quite beneficial Mythopedia (2022).

4.1.2 Dataset Composition

The five distinct categories of mythological animals in the dataset are the centaur, pegasus, hydra, minotaur, and griffin. There were 20 photos in each class, for a total of 100 photos in the dataset. Despite the relatively small number, it accurately illustrates the challenges associated with collecting high-quality, specific images for mythological characters. The tiny dataset size provided significant challenges for model training, as will be discussed in the following sections. Each image was manually examined to ensure that it accurately depicted the fabled creature and satisfied the necessary requirements for clarity and resolution. Images that were unclear or of poor quality were removed to maintain the dataset's integrity.

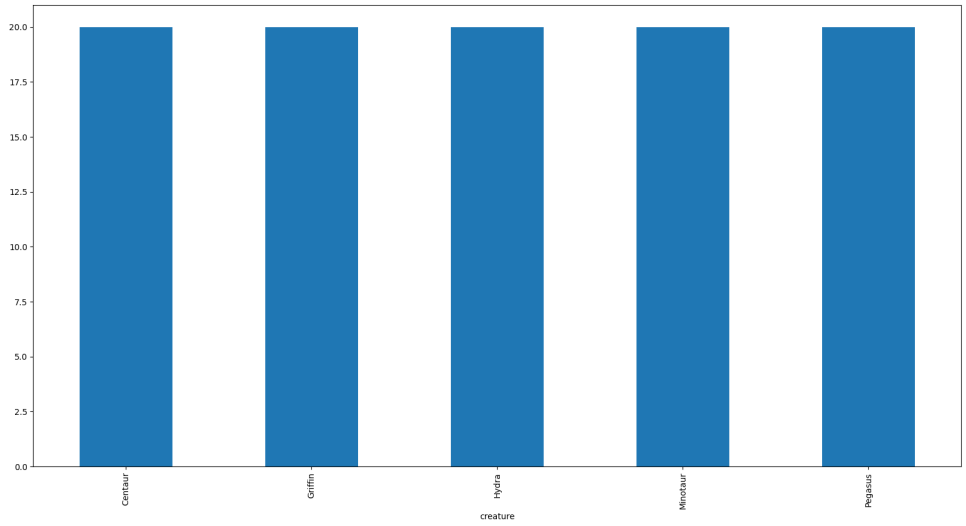


Figure 4.1: Distribution of images across classes

4.1.3 Image Annotation for YOLOv8

In order to pinpoint specific sites connected to each mythological monster, the gathered photographs were labelled for the YOLOv8 object detection model. **Roboflow**, a popular program that makes image annotation simple with an intuitive user interface, was employed for the analysis. The steps involved in the annotation process were as follows:

- **Bounding Box Creation:** A bounding box was carefully drawn around each image to ensure that all relevant parts of the mythical creature were included. In order to train YOLOv8, which requires precise object localisation to accomplish effective detection, this step was crucial Roboflow (2024).
- **Label Assignment:** Every bounding box was given a label with the proper class—such as "Centaur" or "Pegasus"—on it. The labels demonstrated coherence with the CNN model's categories, ensuring coherence between different models Format (2024).
- **Export in COCO Format:** After finished, the annotated dataset was exported in a format called **COCO (Common Objects in Context)**, which is meant to work with YOLOv8. This format includes the image files as well as a JSON file with the annotations (labels and bounding box coordinates, for example). This makes it possible to integrate with the YOLOv8 training pipeline seamlessly Redmon et al. (2015).

4.1.4 Dataset Structuring for CNN

The directory format in which the dataset was organised proved ideal for CNN model picture categorisation tasks. The photos were arranged into folders based on the categories to which they were assigned. The images in each directory featured a single mythological creature and were labelled sequentially (e.g., "Centaur_1.jpg," "Centaur_2.jpg"). Throughout the training phase, this structure was essential for the model to accurately match input photographs with their corresponding labels Sandler et al. (2018).

- **Training and Validation Split:** To account for the dataset's restricted size, the images were divided into distinct training and validation sets. The typical allocation of 80% for training and 20% for validation was used, resulting in 16 training and 4

validation photos for each class. The purpose of choosing this split was to maximise training data and allow for model validation. Szegedy et al. (2015).

- **Data Preprocessing:** Each image's dimensions were changed to 224 by 224 pixels, which is the right input size for the CNN's MobileNetV2 model. Moreover, to ensure consistent input to the model, the pixel values were normalised to the interval [0, 1]. As mentioned in Section 3.3.2 Howard et al. (2017), early testing indicated that augmentation led to overfitting; hence, at this point, no additional augmentation was used.

4.1.5 Challenges Limitations

One of the biggest challenges encountered throughout the data collection process was the limited availability of high-resolution images for certain mythical creatures, particularly those that are under-represented in contemporary media. The restricted supply of specialist datasets poses an intrinsic difficulty to deep learning, impeding the model's ability to be generalised. Due to YOLOv8's performance constraints, the annotations had little effect on the model's final growth. Due to this constraint, the model's complexity and training have to be approached carefully; this is covered in greater detail in later sections of this chapter Goodfellow, Bengio and Courville (2016).

4.2 Model Selection

4.2.1 Initial Model Selection: YOLOv8 for Object Detection

The chosen initial model for this work was YOLOv8, an established model known for its exceptional effectiveness in real-time object detection Redmon et al. (2015). The exceptional speed and accuracy of YOLOv8 rendered it very suitable for the given task. The dataset comprising legendary creatures was annotated in the COCO format, which is inherently compatible with the presented model. Yet, despite efforts to fine-tune hyperparameters and augment data, YOLOv8 failed to detect any animals during the testing period. The insufficient performance of the model may be ascribed to the restricted dataset including 100 images assigned to five categories, together with the unique attributes of the animals, which significantly differ from the standard COCO items. To address these challenges, the study shifted to a more suitable approach for image categorisation by using CNNs, which are particularly suitable for tiny, specialised datasets.

4.2.2 Final Model Selection: MobileNetV2 for Image Classification

Upon realising that YOLOv8 had limitations, the research switched to MobileNetV2, a CNN architecture that is well-known for its efficiency and ability to draw conclusions from small datasets. MobileNetV2 lowers computing complexity without sacrificing accuracy by utilising linear bottlenecks and inverted residuals Sandler et al. (2018).

To address the issue of overfitting that was noted in increasingly complex models, the choice was made to switch to MobileNetV2. MobileNetV2 was better suited for fine-tuning with fewer parameters and successfully handling image classification problems than YOLOv8, which struggled with the small and specialised dataset.

On the basis of its remarkable accuracy and capacity for generalisation on the dataset of mythical animals, MobileNetV2 (Hub, 2020) was chosen as the final model. The research goal of creating a very effective picture classification model for legendary animals was effectively met by this choice, which also solved the issues with YOLOv8.

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
hub_layer (HubLayer)	(None, 1280)	0
batch_normalization (BatchNormalization)	(None, 1280)	5,120
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 512)	655,872
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 5)	2,565

Total params: 663,557 (2.53 MB)
Trainable params: 660,997 (2.52 MB)
Non-trainable params: 2,560 (10.00 KB)

Figure 4.2: Summary of the customised MobileNetV2 model used for mythological creature classification

4.2.3 Justification for Model Transition

A significant turning point in the study was the transition from YOLOv8 to MobileNetV2, which showed a change in technique to address the issues found. The failure of YOLOv8 demonstrated how important it is to properly select a model that complements the special features of the dataset. When used on highly specialised and limited datasets, advanced models designed for large-scale item detection might not always yield adequate results. Despite initial challenges, MobileNetV2 proved to be a more appropriate model for this work because of its focus on efficient photo categorisation, which allowed the research to achieve its objectives.

4.3 Model Architecture

An extensive explanation of the CNN architecture based on MobileNetV2-classifies legendary entities in this section. The architecture's goal was to employ MobileNetV2 for transfer learning by incorporating extra layers and regularisation techniques to help the model fit the given dataset.

4.3.1 Initial Model Architecture and Hyperparameter Tuning

The initial CNN model was designed with the following architecture and hyperparameters:

- **Input Shape: (224, 224, 3)** Images were resized to 224 x 224 pixels to meet the input requirements of MobileNetV2.
- **Output Shape:** Number of classes, which is 5 (Centaur, Griffin, Minotaur, Pegasus, Hydra).

The hyperparameters used in the initial model are summarised in Table 3.1, were taken from taken from below:

Hyperparameter	Values Considered
Optimiser	SGD, RMSProp, Adam
Activation Function	Leaky ReLU, PReLU, ReLU
Learning Rate	1e-5, 1e-4, 1e-3, 1e-2, 0.1
Batch Size	32, 64, 128, 256, 512
Dropout Rate	0.25, 0.5

Table 4.1: Hyperparameters and their ranges for initial model tuning Nazir, Patel and Patel (2018).

The function used to build the model is as follows:

```
def create_model_tf(input_shape=INPUT_SHAPE,
    output_shape=OUTPUT_SHAPE, model_url=MODEL_URL,
                        optimizer='SGD', activation='leaky_relu',
                            learning_rate=1e-3,
                        dropout_rate=0.5, l1_reg=0.05, l2_reg=0.05):
     pass
```

Listing 4.1: Python code for creating a TensorFlow CNN model

The optimal combination was identified by evaluating several activation functions and optimisers within this initial configuration. The implementation of adaptive regularisation involved the use of L1 and L2 penalties, and dropout was employed to reduce overfitting.

4.3.2 Issues with Overfitting

Throughout the training phase, the model displayed substantial overfitting, mainly caused by the following factors:

- **Model Complexity:** A comprehensive hyperparameter tuning process, including the inclusion of many activation functions, optimisers, and regularisation methods, increased the complexity of the model. The complex architecture of the model made it prone to overfitting, especially when working with a small dataset of 100 images classified into 5 categories Sandler et al. (2018).
- **Detecting Overfitting:** Although the model demonstrated a high degree of accuracy on the training data, its performance on the validation data was poor, indicating that it had developed the capacity to memorise the training instances rather than effectively generalise to new data Howard et al. (2017).
- **Data Augmentation Impact:** The dataset was initially augmented using a comprehensive data augmentation technique in order to increase its diversity. Moreover, this exacerbated the issue of overfitting by introducing random fluctuations and superfluous noise into the training data, leading to insufficient generalisation Hub (2020).

The fundamental architecture is MobileNetV2, a condensed CNN intended to attain accuracy and efficiency at the same time. In order to limit computational resource consumption

without sacrificing performance, the MobileNetV2 model makes use of linear bottlenecks and inverted residuals Sandler et al. (2018); Howard et al. (2017). The suggested model makes use of a number of depthwise separable convolutions to lessen the computing load in comparison to conventional convolutional layers. Because of this feature, it works particularly effectively in situations with constrained computing resources and small datasets. Appendix B.1 contains comprehensive information about the methods used for activation extraction, picture preprocessing, and visualisation.

4.3.3 Customisation for Mythological Creature Classification

The MobileNetV2 model was designed to precisely meet the needs of categorising legendary creatures. The changes that stand out include:

- **Feature Extraction Layer:** The TensorFlow Hub framework provides a pre-trained MobileNetV2 feature extractor that is used in the approach as an alternative. This pre-trained layer processes the input photos to create a 1280-dimensional feature vector Hub (2020).
- **Dropout Layer:** The inclusion of a dropout layer with a dropout rate of 0.5 was designed to reduce the chance of overfitting, especially considering the limited size of the dataset. The dropout technique randomly assigns a portion of input units to zero during training, thereby reducing the potential for the model to become too reliant on any single feature Srivastava et al. (2014).
- **Batch Normalisation:** Batch normalisation was put in place after implementing the dropout layer. The purpose of this layer is to standardise the activations of the previous layer, promote stability, and expedite the training process by ensuring uniform data distributions across the network Szegedy et al. (2015).
- **Dense Output Layer:** The newest layer has a compact stratum that includes five components, representing the five classes of mythical creatures (Centaur, Griffin, Minotaur, Pegasus, and Hydra) A softmax activation function is used in this layer to generate class probabilities as output. The weights of this layer were adjusted using L2 regularisation with a regularisation strength of 0.01 to address overfitting more effectively Goodfellow, Bengio and Courville (2016).

4.3.4 Model Implementation

The complete model implementation is defined in the create_model_tf function. Key aspects of the model architecture are summarised in 4.2.

```
def create_model_tf(input_shape=INPUT_SHAPE,
    output_shape=OUTPUT_SHAPE, model_url=MODEL_URL):
     base_model = HubLayer(model_url, trainable=True)

     inputs = tf.keras.Input(shape=input_shape)
     x = base_model(inputs)

     x = tf.keras.layers.BatchNormalization()(x)
     x = tf.keras.layers.Dropout(0.5)(x)
     x = tf.keras.layers.Dense(512, activation='relu',
         kernel_regularizer=l2(0.01))(x)
```

```
    x = tf.keras.layers.Dropout(0.5)(x)

    outputs = tf.keras.layers.Dense(output_shape,
        activation="softmax")(x)

    model = tf.keras.Model(inputs, outputs)

    model.compile(
        optimizer=tf.keras.optimizers.Adam(),
        loss='categorical_crossentropy',
        metrics=['accuracy']
    )
    return model
```

Listing 4.2: Python code for TensorFlow CNN model

4.3.5 Training and Regularisation

In order to enhance performance and prevent overfitting, the model training involves several key components:

- **ReduceLROnPlateau:** A learning rate scheduler that is specialised in its ability to decrease the learning rate at a point when a metric no longer improves. In the event that there is no improvement after a set of epochs (patience), the learning rate is reduced by a factor based on the monitored parameter (like validation loss). It is particularly advantageous when the model is stagnating, as it can aid in transcending local minima and sustaining the learning process Al-Kababji, Bensaali and Dakua (2022).
- **Early Stopping:** The validation loss is monitored by a preemptive callback that stops training when it no longer improves, preventing overfitting and guaranteeing the preservation of the optimal model weights Prechelt (1998).

  The train_model function is where the training procedure is contained, and it uses callbacks to improve the model's performance.

```
def train_model():
    model = create_model_tf()

    tensorboard = create_tensorboard_callback()

    history = model.fit(
        train_data,
        epochs=NUM_EPOCHS,
        validation_data=val_data,
        validation_freq=1,
        callbacks=[tensorboard, early_stopping, reduce_lr]
    )

    return model, history
```

Listing 4.3: Python code for training the CNN model

This architecture and training program was created specifically to address the unique challenges that arise when classifying mythological creatures. Both transfer learning and custom modifications are used to achieve efficient classification performance.

4.4 Training the Model

The methods used to train CNN models are comprehensively covered in this subsection. It encompasses precise training parameters, data augmentation techniques, and approaches to solve problems like overfitting. Furthermore, it provides details on the evaluation of the model's performance.

4.4.1 Training Parameters

The training of the CNN model required the specification of several essential parameters in 4.2. The parameters selected were determined by the model's architecture and the dataset's specifications.

Parameter	Value
Epochs	100
Batch Size	32
Optimizer	Adam
Learning Rate	1e-3
Activation	ReLU
Dropout Rate	0.5
Loss Function	Categorical Crossentropy
Metrics	Accuracy
Early Stopping	Patience = 10 epochs
Learning Rate Scheduler	Exponential decay with a factor of 0.1

Table 4.2: Training parameters used for the final CNN model.

4.4.2 Data Augmentation

To increase the model's resilience and diversity, data augmentation was used to expand the training dataset. The initial method involved a comprehensive augmentation technique that involved random flipping, brightness modulations, and cropping. In order to decrease overfitting, a more selective augmentation approach was utilised for the ultimate model.

Augmentation Techniques Used:

- **Random Horizontal Flip:** By flipping photos horizontally, different perspectives can be replicated.
- **Random Vertical Flip:** To enhance dataset diversity, perform vertical image flipping.
- **Random Brightness:** Adjusting image brightness to deal with fluctuation in lighting conditions.
- **Random Contrast:** Various situations can be simulated through the modification of image contrast.

- **Random Saturation:** Adjusting the saturation of colors to make them more vibrant.
- **Random Hue:** Adapting the visual hue to fit diverse color distributions.
- **Central Cropping:** By trimming the middle of the image, the effect of focusing on various sections of the object can be replicated.

Figure 4.3: An illustration of how to apply data augmentation techniques to images.

4.4.3 Training Procedure

The model was trained using TensorFlow and Keras libraries. The training process involved the following steps:

1. **Data Preparation:**
   - **Training Data:** To avoid overfitting, images were preprocessed and grouped into batches using the create_data_batches functionChollet (2017).
   - **Validation Data:** To ensure consistency, validation data was created in separate batches without any shuffling Sarker (2021).
2. **Model Compilation:**
   - The Adam optimiser was used to develop the model, which has a learning rate of 1e-3 and a loss function that utilises Categorical Crossentropy. The use of this strategy is particularly effective in resolving multi-class classification problems Kingma and Ba (2014).
3. **Callbacks:**
   - **Early Stopping:** To prevent overfitting and reduce computing expenses, it is necessary to stop training if the validation loss doesn't increase for 10 epochs

Prechelt (1998).

- **ReduceLROnPlateau:** The learning rate was dynamically adjusted using an adaptive ReduceLROnPlateau method. By decreasing the learning rate as the validation loss reaches its maximum improvement, this method can enable more precise convergence towards the end of the training phase.

4. **Training Execution:**
   - The model was trained for a maximum of 100 epochs, with the ability to pause training immediately if needed. To ensure convergence and prevent overfitting, the training history was closely watched. Smith (2015).

Figure 4.4: The accuracy and loss curves for training and validation over time.

4.5 Evaluation Metrics

Understanding the effectiveness of the CNN model in categorising mythological creature photos requires an evaluation of its performance. The following section gives an overview of the metrics used to evaluate the model, which include accuracy, precision, recall, F1-score, and the confusion matrix Davis and Goadrich (2006); Wei (2024). A thorough assessment requires valuable insights into various aspects of model performance, which are provided by the many metrics.

Appendix A provides more in-depth information on evaluation metrics such as accuracy, precision, recall, F1 score, and overall score.

4.5.1 Accuracy

Accuracy is the ratio between the number of instances that were accurately classified and the total number of observed instances. A thorough understanding of the model's performance is provided by this statistic, which is essential. In this study, accuracy is defined as:

$$Accuracy = \frac{NumberofCorrectPredictions}{TotalNumberofPredictions}$$

The final model's accuracy was 70%, which means that it correctly categorised 70% of the photos in the test set.

4.5.2 Precision

Precision is defined as the percentage of positive observations that are accurately predicted from all the expected positive samples. The accuracy of optimistic forecasts is quantified by this metric. In situations where the consequences of incorrect positive results are significant, precision is especially crucial. Precision for each class is calculated as follows:

$$Precision = \frac{TP}{TP + FP}$$

where TP is the number of true positives, and FP is the number of false positives.

Macro Precision is the average precision score calculated for each class. The model that was created achieved a macro precision of 72.33%, which indicates that it is effective in categorising various mythological creatures.

4.5.3 Recall

Recall, also known as sensitivity, is the ratio of positively predicted observations among all the observations in the appropriate class. The model's ability to detect positive events is measured by this metric. Recall is calculated as:

$$Recall = \frac{TP}{TP + FN}$$

where FN is the number of false negatives.

Macro Recall is the average recall level for all classes. The importance of this statistic stems from the significant cost of false negatives and its ability to quantify the model's ability to accurately identify all relevant instances.

4.5.4 F1 Score

The **F1 Score** is calculated using the harmonic mean of precision and recall. The goal is to achieve a balanced mix of precision and recall. The F1 score is calculated as:

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Macro F1 Score generates the average F1 score for every class, enabling a complete assessment of the model's performance in situations with potential class imbalance.

A well-balanced performance across several classes was confirmed by the model's macro F1 score of 70.29%.

4.5.5 Confusion Matrix

A confusion matrix is a tabular presentation that compares the expected and actual class labels to show the classification model's performance. This analysis provides valuable information

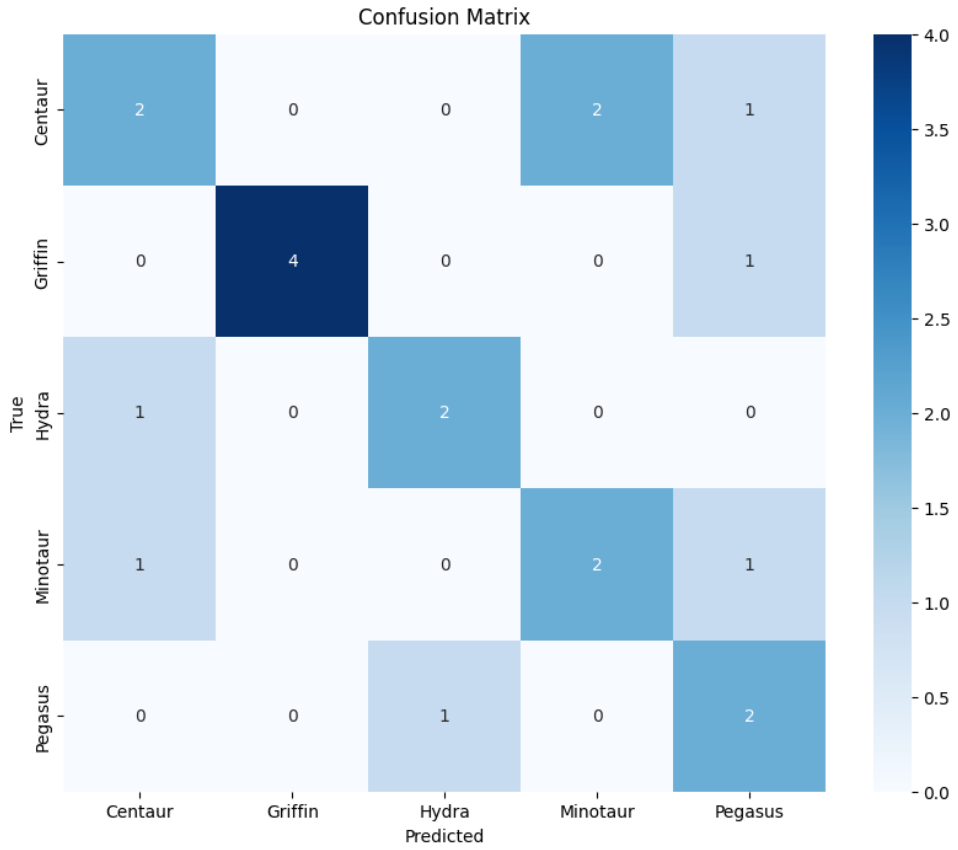


Figure 4.5: CNN Confusion Metrix

about the specific categories of errors produced by the model, which include both false positives and false negatives Grandini, Bagli and Visani (2020).

By analysing the confusion matrix, it can be concluded that the model is highly effective in correctly identifying Hydra and Minotaur with a high recall rate. The model is facing difficulties in correctly identifying Griffin and Pegasus, which suggests potential areas for model enhancement.

4.5.6 Classification Report

A comprehensive review of model performance is given by the Classification Report by combining precision, recall, and F1-score for each class Sokolova and Lapalme (2009). The report for the final model is as follows:

Metric	Value
Accuracy	0.6000
Precision (Macro)	0.6133
F1 Score (Macro)	0.6000

Table 4.3: Overall performance metrics for the CNN model.

Class	Precision	Recall	F1-Score	Support
Centaur	0.50	0.40	0.44	5
Griffin	1.00	0.80	0.89	5
Hydra	0.67	0.67	0.67	3
Minotaur	0.50	0.50	0.50	4
Pegasus	0.40	0.67	0.50	3
Accuracy	0.60 (on 20 samples)			
Macro avg	0.61	0.61	0.60	20
Weighted avg	0.64	0.60	0.61	20

Table 4.4: Classification report for the mythological creatures dataset.

Chapter 5

Results and Discussion

5.1 Model Performance

The performance of two models is compared in this subsection: the final selected model and a more detailed model, which was selected after hyperparameter adjustment. The evaluation of each model involved the use of training and validation accuracy, loss curves, confusion matrices, and important assessment metrics like precision, recall, and F1-score. The main purpose is to determine if the model is better able to generalise to unfamiliar data while maintaining balanced classification performance.

5.1.1 Training and Validation Accuracy

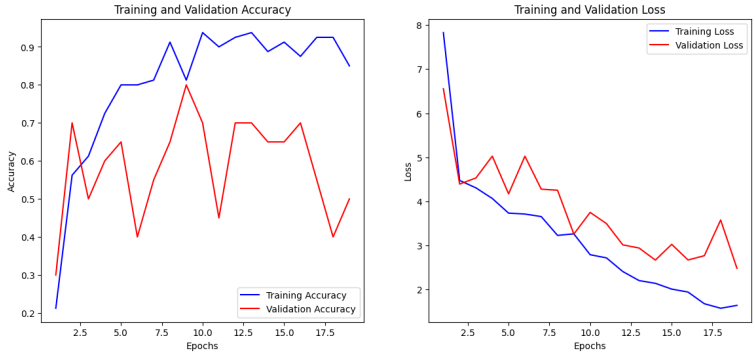
The training and validation accuracy for the final model are shown in Figure 5.1 (a), whereas the complicated model achieves the same accuracy in Figure 5.1 (b). An approximate training accuracy of 95% and a final validation accuracy of 60% were attained by the final model. Notwithstanding the variability in the validation accuracy, the model consistently performed, suggesting that it generalised rather well despite the limited dataset.

Conversely, the intricate model (Figure 5.1, a) exhibited superior training accuracy, exceeding 90%, but its validation accuracy was more inconsistent, concluding at 55%. The substantial discrepancy between the training and validation accuracies indicates that the intricate model experienced overfitting, rendering it less appropriate for this application in comparison to the simpler option Caruana and Niculescu-Mizil (2006); Ng (2004).

5.1.2 Training and Validation Loss

The loss curves shown in figures 5.1 (a) and 5.1 (b) serve to emphasise the substantial disparities between the two models. The final model exhibited a consistent decrease in both the training and validation loss. However, the validation loss reached a plateau after a few epochs, suggesting a limited degree of overfitting. Nevertheless, the impact of this phenomenon was minimal compared to the complex model, which exhibited a swift decrease in training loss but inconsistent validation loss. This observation suggests that overfitting is the primary problem with the complex model Srivastava et al. (2014).

(a) Final Model



(b) HyperParams Model

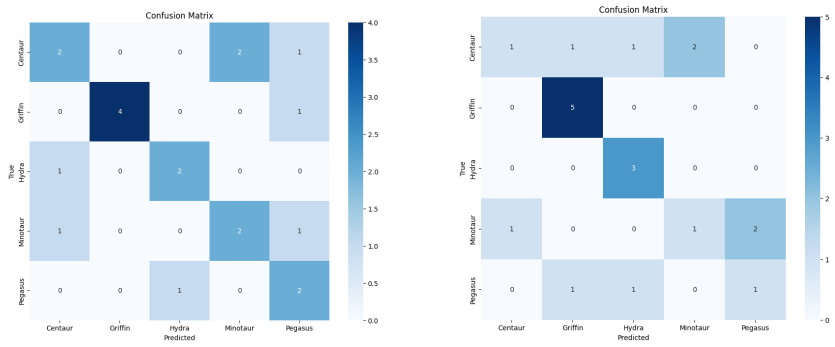
Figure 5.1: Loss and Accuracy curve for both the models

5.1.3 Confusion Matrix

The confusion matrices for the basic and complex models are shown in Figures 5.2(a) and 5.2(b), respectively. Both models demonstrated accurate categorisation of **Griffin** and **Hydra**, with **Griffin** attaining perfect classification in both models. Nevertheless, categorising **Centaur** and **Pegasus** proved to be more difficult for both models, as they often yielded incorrect classifications. As shown in Figure 5.2 (a), the final model incorrectly classified **Centaur** as **Griffin** and **Pegasus**. Conversely, the complex model (Figure 5.2 (b)) misclassified **Centaur** more often, providing further evidence that the simpler model was more dependable in generalising to unfamiliar data.

5.1.4 Evaluation Metrics

Table 5.1 displays the analytical measures used to evaluate both models. The final model attained a superior overall accuracy of 60% and F1-score of 0.60 in comparison to the complex model which earned an accuracy of 55% and an F1-score of 0.50. The basic architecture was shown to be more balanced and successful across all classes. Notwithstanding its increased number of layers and superior training accuracy, the intricate model encountered difficulties in generalising and sustaining consistent performance across all categories Simard, Steinkraus and Platt (2003).



(a) Final Model (b) HyperParams Model

Figure 5.2: Confusion Matrix

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
Final Model	60%	0.6133	0.6000	0.6000
Complex Model	55%	0.4962	0.5600	0.4976

Table 5.1: Summary of Evaluation Metrics for Final and Complex Models

5.1.5 Overfitting and Generalisation

Both models exhibited different levels of overfitting, with the complicated model in particular displaying a greater degree of overfitting in comparison to the final model. The sophisticated model achieved a training accuracy of around 90%, but its validation accuracy varied greatly, indicating severe limitations in its ability to generalise to new data Yosinski et al. (2014). By comparison, the ultimate model, albeit attaining a lower level of training accuracy, had a more consistent performance on the validation set, suggesting superior generalisation He et al. (2016).

Therefore, the decision to utilise the simpler model was confirmed by its capacity to maintain a balance between training and validation performance. This finding aligns with previous studies that suggest shorter models tend to exhibit higher generalisation capabilities, especially when the training data is scarce Zhang et al. (2021). Moreover, simpler models exhibit greater computational efficiency, rendering them well-suited for practical applications that have restricted processing capabilities Howard et al. (2017). Considering the intricate nature of the assignment regarding the classification of mythological creatures, the simpler model shown not only superior generalisation but also more effective utilisation of resources, therefore establishing itself as the superior option for this challenge.

5.2 Impact of Data Augmentation

Implementing data augmentation is essential for improving the generalisation of models, particularly when dealing with datasets that are limited or imbalanced. This section provides an overview of the influence of data augmentation on both basic and sophisticated models, encompassing the many types employed and their consequences on the performance of the models in classifying mythological fauna.

5.2.1 Types of Data Augmentation Applied

Conventional methods such as random rotations, flipping, zooming, and brightness modifications were employed to promote diversity in the training dataset and avoid overfitting Krizhevsky, Sutskever and Hinton (2012). These augmentations facilitated the simulation of real-world variations and enhanced the models' capacity to generalise by exposing them to various characteristics of each class. This was especially beneficial for differentiating related classes such as **Centaur** and **Pegasus**.

5.2.2 Reduced Overfitting

The implementation of augmentation resulted in a decrease in overfitting, as seen by the more consistent validation curves and reduced validation loss for the complicated model. This is consistent with studies demonstrating that data augmentation improves resilience and decreases overfitting by generating a more diverse training dataset Shorten and Khoshgoftaar (2019); Perez and Wang (2017).

5.2.3 Considerations for Future Work

Potential future research might investigate sophisticated augmentation methods such as CutMix Yun et al. (2019) or Mixup Zhang et al. (2021), as well as generative models like GANs Goodfellow, Bengio and Courville (2016) to further boost dataset diversity. Nevertheless, it is imperative to refrain from excessive augmentation, since it can potentially introduce noise and diminish the performance of the model Lemley, Bazrafkan and Corcoran (2017). Adopting a balanced augmentation strategy is crucial for attaining maximum model enhancements.

5.3 Discussion on Results

This part assesses the advantages and constraints of the basic and advanced models, and suggests ways to enrich the categorisation of mythological creatures.

5.3.1 Strengths of the Final Model

The basic model demonstrated exceptional generalisation and computational efficiency, attaining a validation accuracy of 60% and a consistent learning curve with minimal overfitting. The simplicity of the system was beneficial considering the little dataset, as it effectively balanced accuracy and resource utilisation, making it well-suited for real-time applications Caruana and Niculescu-Mizil (2006); Chollet (2017); Howard et al. (2017).

5.3.2 Strengths of the Complex Model

The intricate model exhibited superior training accuracy (around 90%) and the capacity for comprehensive feature acquisition, which might be enhanced with a sizeable dataset. Especially beneficial for applications that demand exceptional accuracy, such as academic research He et al. (2016). Further refinement and data analysis may enable it to exceed the performance of the basic model.

5.3.3 Limitations of the Models

Both models exhibited constraints: the basic model had difficulties in handling complex class structures because of its superficial design, resulting in misclassifications within comparable classes such as **Centaur** and **Pegasus** Simard, Steinkraus and Platt (2003). The sophisticated model encountered overfitting problems, exhibiting strong performance on training data but performing inadequately on unfamiliar data, underscoring the importance of meticulous model selection and tweaking Yosinski et al. (2014).

5.3.4 Areas for Improvement

To optimise model performance, examine the following:

1. **Data Augmentation and Expansion:** Generate a larger dataset and apply advanced augmentation techniques to improve generalisation Yun et al. (2019). According to Buda, Maki and Mazurowski (2017); Perez and Wang (2017) by implementing data augmentation, the number of under-represented classes was increased, therefore achieving a balanced dataset and ultimately improving the performance and accuracy of the model.
2. **Regularisation Techniques:** Application of dropout, batch normalisation, and weight decay techniques is proposed to mitigate overfitting Srivastava et al. (2014).
3. **Hyperparameter Tuning:** Fine-tune hyperparameters and employ sophisticated optimisation methods to enhance performance Bergstra and Bengio (2012).
4. **Transfer Learning:** Augment accuracy by employing pre-trained models such as EfficientNet or ResNet Tan and Le (2019).
5. **Interpretable Models:** Utilise interpretability techniques such as Grad-CAM or SHAP to comprehend and enhance model reasoning Ribeiro, Singh and Guestrin (2016).

5.3.5 Conclusion on Model Selection

The current practicality of the simple approach for this task is attributed to its generalisation and superior efficiency. Nevertheless, by including more data and enhancing experimental techniques, the intricate model has the potential to outperform, especially for tasks that demand accurate categorisation.

Chapter 6

Conclusions

6.1 Summary of Findings

This dissertation investigated CNNs for the categorisation of legendary creatures. The shift from YOLOv8 to MobileNetV2 was made given the superior suitability of MobileNetV2 for image classification tasks.

Primary discoveries comprise:

1. **Model Transition:** The YOLOv8 model, designed for handling extensive datasets, showed reduced efficacy when applied to the smaller and more specialised dataset. The classification-optimised MobileNetV2 demonstrated superior efficiency, underscoring the significance of matching model selection with task demands.
2. **Overfitting and Regularisation:** The initial models encountered overfitting problems as a result of constraints in the dataset. Advanced methods such as dropout and L2 regularisation enhanced the generalisation of the model, resulting in a final model that successfully balanced complexity and regularisation.
3. **Performance Metrics:** Ultimately, the MobileNetV2 model attained a 60% accuracy and a 60% macro-average F1 score. Although Centaur and Minotaur were identified with greater precision, Pegasus and Griffin presented distinct difficulties. Higher misclassifications were seen among visually comparable classes, suggesting the necessity for dataset augmentation.
4. **Dataset Size Challenges:** The restricted dataset of 100 photos hindered the ability to generalise. The early exacerbation of overfitting by extensive data augmentation highlights the significance of dataset size and variety in achieving accuracy and generalisation.
5. **Model Evaluation and Comparison:** MobileNetV2 surpassed YOLOv8 in terms of both accuracy and generalisation precision. This work validates that less complex models, when properly regularised, tend to be more efficient when applied to small, specialised datasets.

The observed outcomes underscore the importance of model selection, regularisation, and dataset properties in the field of deep learning. This work showcases the capacity of CNNs to classify mythological creatures and establishes a foundation for further investigation in specialised classification tasks.

6.2 Contributions to the Field

This dissertation provides significant advances to the field of picture classification, namely, deep learning applied to the classification of mythological creatures. Primary contributions comprise:

1. **CNNs for Mythological Creatures:** Among the first to use CNNs, namely MobileNetV2, to categorise mythological creatures, this study demonstrates that deep learning may be tailored for culturally important and specialised fields.
2. **Model Transition and Optimisation:** This work emphasises the need of adjusting the complexity of models to match the size of the dataset. It shows that transitioning from YOLOv8 to MobileNetV2 enhanced performance for small, specialised datasets.
3. **Overfitting Mitigation Insights:** This study offers significant insights into the management of overfitting using methods such as dropout and L2 regularisation. It also tackles the limited efficacy of extensive data augmentation for small datasets.
4. **Custom Dataset Development:** Development of an innovative dataset of mythological creatures establishes a standard for future study and emphasises the difficulties and approaches for handling unconventional datasets.
5. **Evaluation Metrics and Comparison:** Systematic assessment of performance utilising accuracy, precision, recall, and F1-score, together with model comparisons, provides a benchmark for comparable specialised classification tasks.
6. **Specialised Classification Tasks:** This work expands the use of CNNs to cultural and historical realms, therefore showcasing the potential of deep learning in areas as peoplelore and archaeology.

In general, this study represents progress in deep learning techniques for certain fields, providing useful answers and creating new opportunities for further investigation in specialised image categorisation.

6.3 Limitations of the Study

This study faced several limitations impacting its scope and results:

1. **Dataset Size and Quality:** The low dataset of 100 images hindered the performance and generalisation of the model, worsened by the lack of diversity in image circumstances and the possible biases resulting from manual retrieval.
2. **Model Complexity and Overfitting:** Initial sophisticated models exhibited overfitting to the training data. While simplifications and regularisation were beneficial, they limited the model's capacity to acquire intricate characteristics, hence emphasising a compromise between model complexity and scale of the dataset.
3. **Generalisability:** The generalisability of the approach to other datasets or artistic iterations of mythological beings is yet uncertain. The study's concentration on a limited and tailored dataset restricts its relevance to wider or distinct specialised jobs.
4. **Lack of Comparison with Advanced Models:** The study did not investigate more sophisticated architectures such as EfficientNet or deeper implementations of ResNet,

which could have enhanced the outcomes but were constrained by time and resources.

5. **Limited Hyperparameter Tuning:** Numerical resources limited the grid search for hyperparameter adjustment. Further optimisation of model performance may have been achieved by a more comprehensive search.

6. **Class Imbalance:** Although attempts were made to achieve a balanced dataset, certain classes were more accurately identified than others because of the presence of overlapping visual characteristics, which highlights the difficulties associated with limited datasets.

To overcome these constraints, one could improve model performance and generalisability by expanding the dataset, using sophisticated models, and fine-tuning hyperparameter estimates. This discovery establishes a fundamental basis for future investigations in deep learning for challenges involving specialised classification.

6.4 Future Work

This study has many opportunities for further investigation, namely in overcoming the constraints of dataset size, model complexity, and the applicability of the categorisation model for mythological fauna. Primary areas of focus for future research encompass:

1. **Dataset Expansion**
   Augmenting the dataset is essential for enhancing model performance. In order to accomplish this, it is necessary to augment the quantity of images, include additional mythical categories, and integrate a wider range of styles, time periods, and depictions. Employing Generative Adversarial Networks (GANs) for synthetic data generation has the potential to augment dataset size without requiring manual gathering Goodfellow, Bengio and Courville (2016).

2. **Exploring Advanced Architectures**
   In order to enhance performance, future research should investigate more sophisticated architectures such as EfficientNet and ResNet. These architectural designs have the potential to provide enhanced scalability and accuracy, especially when trained on greater datasets He et al. (2016); Tan and Le (2019). Furthermore, Vision Transformers (ViTs) are highly suitable for capturing the whole context in intricate mythical imagery Dosovitskiy et al. (2020).

3. **Application in Real-World Scenarios**
   The model can be applied to real-world tasks, such as:

   (a) **Cultural Heritage and Museums:** Providing support in the catalogue and categorisation of mythological creatures found in artefacts and artworks.

   (b) **Augmented Reality (AR) and Gaming:** Optimising user engagement through real-time identification of mythological animals Azuma (1997).

   (c) **Education and Storytelling:** Promotion of education and conservation of cultural heritage using interactive technologies and applications Dede (2009).

4. **Extending to Other Non-Standard Domains**
   Further expansion of the methodology to include the categorisation of other non-standard images, such as folklore creatures, historical artefacts, or cultural iconography, would

enhance the significance of this research in fields such as archaeology, anthropology, and cultural studies Hollink et al. (2004).

5. **Improving Model Interpretability:**
   Potential future research might prioritise the improvement of model interpretability by employing methods such as **Grad-CAM** or saliency maps. These approaches would offer more comprehensive understanding of the decision-making process of the model, which is crucial for specialised applications Selvaraju et al. (2016).

   In brief, the performance and impact of the model can be greatly improved by increasing the dataset, investigating more sophisticated designs, and implementing it in real-world scenarios. This will enable its application to a wider array of specialised picture classification jobs.

Bibliography

Al-Kababji, A., Bensaali, F. and Dakua, S.P., 2022. Scheduling techniques for liver segmentation: Reducelronplateau vs onecyclelr [Online]. 2202.06373, Available from: https://arxiv.org/abs/2202.06373.

Azuma, R.T., 1997. A survey of augmented reality. *Presence: Teleoper. virtual environ.* [Online], 6(4), p.355–385. Available from: https://doi.org/10.1162/pres.1997.6.4.355.

Bergstra, J. and Bengio, Y., 2012. Random search for hyper-parameter optimization. *J. mach. learn. res.*, 13(null), p.281–305.

Bojarski, M., Testa, D.D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J. and Zieba, K., 2016. End to end learning for self-driving cars. *Corr* [Online], abs/1604.07316. 1604.07316, Available from: http://arxiv.org/abs/1604.07316.

Buda, M., Maki, A. and Mazurowski, M.A., 2017. A systematic study of the class imbalance problem in convolutional neural networks. *Corr* [Online], abs/1710.05381. 1710.05381, Available from: http://arxiv.org/abs/1710.05381.

Caruana, R. and Niculescu-Mizil, A., 2006. An empirical comparison of supervised learning algorithms [Online]. *Proceedings of the 23rd international conference on machine learning*. New York, NY, USA: Association for Computing Machinery, ICML '06, p.161–168. Available from: https://doi.org/10.1145/1143844.1143865.

Chollet, F., 2017. *Deep learning with python*. 1st ed. USA: Manning Publications Co.

Davis, J. and Goadrich, M., 2006. The relationship between precision-recall and roc curves [Online]. *Proceedings of the 23rd international conference on machine learning*. New York, NY, USA: Association for Computing Machinery, ICML '06, p.233–240. Available from: https://doi.org/10.1145/1143844.1143874.

Dede, C., 2009. enImmersive interfaces for engagement and learning. *Science*, 323(5910), pp.66–69.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. 2010.11929.

Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M. and Thrun, S., 2017. enCorrigendum: Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 546(7660), p.686.

Format, C.D., 2024. Common objects in context (coco). `https://cocodataset.org`. Accessed: 2024-08-01.

Gatys, L.A., Ecker, A.S. and Bethge, M., 2016. Image style transfer using convolutional neural networks [Online]. *2016 ieee conference on computer vision and pattern recognition (cvpr)*. pp.2414–2423. Available from: `https://doi.org/10.1109/CVPR.2016.265`.

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation [Online]. *Proceedings of the 2014 ieee conference on computer vision and pattern recognition*. USA: IEEE Computer Society, CVPR '14, p.580–587. Available from: `https://doi.org/10.1109/CVPR.2014.81`.

Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep learning*. MIT Press. `http://www.deeplearningbook.org`.

Grandini, M., Bagli, E. and Visani, G., 2020. Metrics for multi-class classification: an overview. *Arxiv* [Online], abs/2008.05756. Available from: `https://api.semanticscholar.org/CorpusID:221112671`.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition [Online]. *2016 ieee conference on computer vision and pattern recognition (cvpr)*. pp.770–778. Available from: `https://doi.org/10.1109/CVPR.2016.90`.

Hollink, L., Schreiber, A.T., Wielinga, B.J. and Worring, M., 2004. enClassification of user image descriptions. *Int. j. hum. comput. stud.*, 61(5), pp.601–626.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. `1704.04861`.

Hub, T., 2020. Pre-trained model: Mobilenetv2. `https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4`. Accessed: 2024-08-05.

Karpathy, A., Johnson, J. and Li, F.F., 2024. Cs231n convolutional neural networks for visual recognition. `https://cs231n.github.io/convolutional-networks/`. Accessed: 2024-08-25.

Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. *Corr* [Online], abs/1412.6980. Available from: `https://api.semanticscholar.org/CorpusID:6628106`.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Proceedings of the 25th international conference on neural information processing systems - volume 1*. Red Hook, NY, USA: Curran Associates Inc., NIPS'12, p.1097–1105.

Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the ieee* [Online], 86(11), pp.2278–2324. Available from: `https://doi.org/10.1109/5.726791`.

Lemley, J., Bazrafkan, S. and Corcoran, P., 2017. Smart augmentation - learning an optimal data augmentation strategy. *Corr* [Online], abs/1703.08383. `1703.08383`, Available from: `http://arxiv.org/abs/1703.08383`.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C. and Berg, A.C., 2015. SSD: single shot multibox detector. *Corr* [Online], abs/1512.02325. 1512.02325, Available from: http://arxiv.org/abs/1512.02325.

Lowe, D., 1999. Object recognition from local scale-invariant features [Online]. *Proceedings of the seventh ieee international conference on computer vision*. vol. 2, pp.1150–1157 vol.2. Available from: https://doi.org/10.1109/ICCV.1999.790410.

Mythopedia, 2022. Mythopedia: Encyclopedia of mythology. https://mythopedia.com. Accessed: 2024-08-05.

Nazir, S., Patel, S. and Patel, D., 2018. Hyper parameters selection for image classification in convolutional neural networks [Online]. *17th ieee international conference on cognitive informatics & cognitive computing (icci*cc 2018)*. IEEE, pp.401–407. Available from: https://doi.org/10.1109/ICCI-CC.2018.8482081.

Ng, A.Y., 2004. Feature selection, l1 vs. l2 regularization, and rotational invariance [Online]. *Proceedings of the twenty-first international conference on machine learning*. New York, NY, USA: Association for Computing Machinery, ICML '04, p.78. Available from: https://doi.org/10.1145/1015330.1015435.

Pandey, A., 2018. Depth-wise convolution and depth-wise separable convolution [Online]. [Online; posted 9-September-2018]. Available from: https://medium.com/@zurister/depth-wise-convolution-and-depth-wise-separable-convolution-37346565d4ec/.

Perez, L. and Wang, J., 2017. The effectiveness of data augmentation in image classification using deep learning. *Corr* [Online], abs/1712.04621. 1712.04621, Available from: http://arxiv.org/abs/1712.04621.

Prechelt, L., 1998. Early stopping-but when? *Neural networks: Tricks of the trade, this book is an outgrowth of a 1996 nips workshop*. Berlin, Heidelberg: Springer-Verlag, p.55–69.

Redmon, J., Divvala, S.K., Girshick, R.B. and Farhadi, A., 2015. You only look once: Unified, real-time object detection. *Corr* [Online], abs/1506.02640. 1506.02640, Available from: http://arxiv.org/abs/1506.02640.

Ribeiro, M.T., Singh, S. and Guestrin, C., 2016. "why should I trust you?": Explaining the predictions of any classifier. *Corr* [Online], abs/1602.04938. 1602.04938, Available from: http://arxiv.org/abs/1602.04938.

Roboflow, 2024. Roboflow: The universal developer tool for computer vision. https://roboflow.com. Accessed: 2024-08-01.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks [Online]. *2018 ieee/cvf conference on computer vision and pattern recognition*. pp.4510–4520. Available from: https://doi.org/10.1109/CVPR.2018.00474.

Sarker, I., 2021. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *Sn computer science* [Online], 2, p.420. Available from: https://doi.org/10.1007/s42979-021-00815-1.

Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D. and Batra, D., 2016. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-

based localization. *Corr* [Online], abs/1610.02391. 1610.02391, Available from: http://arxiv.org/abs/1610.02391.

Shorten, C. and Khoshgoftaar, T.M., 2019. enA survey on image data augmentation for deep learning. *J. big data*, 6(1).

Simard, P., Steinkraus, D. and Platt, J., 2003. Best practices for convolutional neural networks applied to visual document analysis [Online]. *Seventh international conference on document analysis and recognition, 2003. proceedings.* pp.958–963. Available from: https://doi.org/10.1109/ICDAR.2003.1227801.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. 1409.1556.

Singh, S., 2022. Top 5 image data augmentation techniques to mitigate overfitting in computer vision [Online]. Accessed: 2024-08-22. Available from: https://www.labellerr.com/blog/top5-image-data-augmentation-techniques-to-mitigate-overfitting-in-computer-vision/

Smith, L.N., 2015. No more pesky learning rate guessing games. *Corr* [Online], abs/1506.01186. 1506.01186, Available from: http://arxiv.org/abs/1506.01186.

Sokolova, M. and Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Inf. process. manage.* [Online], 45(4), p.427–437. Available from: https://doi.org/10.1016/j.ipm.2009.03.002.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. mach. learn. res.*, 15(1), p.1929–1958.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions [Online]. *2015 ieee conference on computer vision and pattern recognition (cvpr)*. pp.1–9. Available from: https://doi.org/10.1109/CVPR.2015.7298594.

Taigman, Y., Yang, M., Ranzato, M. and Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification [Online]. *2014 ieee conference on computer vision and pattern recognition*. pp.1701–1708. Available from: https://doi.org/10.1109/CVPR.2014.220.

Tan, M. and Le, Q.V., 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. 1905.11946.

Wei, D., 2024. Essential math for machine learning: Confusion matrix, accuracy, precision, recall, f1-score [Online]. [Online; posted 28-January-2024]. Available from: https://medium.com/@weidagang/demystifying-precision-and-recall-in-machine-learning-6f756a4c54ac/.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks? *Proceedings of the 27th international conference on neural information processing systems - volume 2*. Cambridge, MA, USA: MIT Press, NIPS'14, p.3320–3328.

Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J. and Yoo, Y., 2019. Cutmix: Regularization

strategy to train strong classifiers with localizable features. *Corr* [Online], abs/1905.04899. 1905.04899, Available from: http://arxiv.org/abs/1905.04899.

Zhang, Z., Xu, S., Zhang, S., Qiao, T. and Cao, S., 2021. enAttention based convolutional recurrent neural network for environmental sound classification. *Neurocomputing*, 453, pp.896–903.

Appendix A

Appendix: Evaluation Metrics

A.1 Accuracy

Definition: The ratio between the number of correct predictions and the total number of predictions made is called accuracy. The overall performance of the model is measured by it.

Mathematical Equation:

$$Accuracy = \frac{NumberofCorrectPredictions}{TotalNumberofPredictions} \tag{A.1}$$

For a binary classification problem, where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

The formula becomes:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{A.2}$$

Derivation: By comparing predicted classifications to actual classifications, the confusion matrix summarises the performance of a classification model, resulting in accuracy.

A.2 Precision

Definition: The ratio of positively predicted observations that are correctly predicted to the total predicted positives is called precision. The model's positive predictions' quality is indicated by it.

Mathematical Equation:

$$Precision = \frac{TP}{TP + FP} \tag{A.3}$$

Derivation: The confusion matrix is the source of precision. The model's predictions are measured based on the proportion of true positive predictions.

A.3 Recall (Sensitivity or True Positive Rate)

Definition: Recall is the percentage of positive observations that were correctly predicted compared to the total number of actual positives. It gauges the model's ability to capture all positive instances.

Mathematical Equation:

$$Recall = \frac{TP}{TP + FN} \tag{A.4}$$

Derivation: The confusion matrix is where recall comes from. The model correctly identified the proportion of actual positives, as indicated.

A.4 F1 Score

Definition: The F1 score is a measurement of the balance between precision and recall. It is useful in cases where the class distribution is unbalanced because it provides a balance between precision and recall.

Mathematical Equation:

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{A.5}$$

Derivation: The F1 score combines accuracy and recall into one metric. In cases of uneven class distributions, it is particularly useful because it accounts for both false positives and false negatives.

A.5 Overall Score

Definition: A composite metric or multiple metrics are often used to evaluate the overall score. It is capable of indicating the model's general performance.

Mathematical Equation: This can be influenced by the context. In a classification context that involves multiple classes, a weighted average of precision, recall, and F1 scores may be utilised for all classes.

$$OverallScore = \frac{1}{N} \sum_{i=1}^{N} Metric_i \tag{A.6}$$

where N is the number of classes and $Metric_i$ can be accuracy, precision, recall, or F1 score for each class.

Derivation: In order to obtain the overall score, performance metrics across different classes can be aggregated or weighted averages can be utilised to reflect the importance of each class in the evaluation.

Appendix B

Appendix: Image Generation and Visualisation

B.1 Image Generation and Visualisation

The process and outcomes of creating and presenting image data from a deep learning model are presented in this appendix. Both the original image and its corresponding feature activations from the model are loaded, preprocessed, and visualised in a series of steps to generate the images.

B.1.1 Loading and Preprocessing the Image

The deep learning model's input requirements are met by loading and preprocessing the image. The image is resized to the appropriate dimensions during preprocessing (224x224 pixels in this case) converting it into an array format and determining the pixel values within the [0, 1] range. The image is prepared to be suitable for input into the model.

B.1.2 Visualising the Original Image and Its RGB Channels

Upon preprocessing the image, it is visualised in both its original form and its individual RGB channels. The visualisation includes:

- **Original Image:** The dataset shows the full-color image.
- **Red Channel:** The red component of the RGB channels is only visible in the image.
- **Green Channel:** The green component of the RGB channels is only visible in the image.
- **Blue Channel:** The blue component of the RGB channels is only visible in the image.

The distribution of color information across different channels can be understood through this.

B.1.3 Visualising Activations of the First Convolutional Layer

The activations of the first convolutional layer are visualised to gain an understanding of the model's internal workings. This layer's convolutional filters produce feature maps through the

activations. The image features are extracted by each filter in different ways.

The visualisation of these activations includes:

- **Feature Maps:** The output of each filter in the first convolutional layer is represented by a grid of images. The layers detect different features, such as edges, textures, or patterns, and each map highlights them.

Usually, the number of feature maps is more than the number of channels in the original image and visualising these maps helps in understanding what specific features are being captured by the model.

B.1.4 Figures

By demonstrating the preprocessing steps and activations of the convolutional layer, these figures shed light on the image data and the model's internal feature extraction.

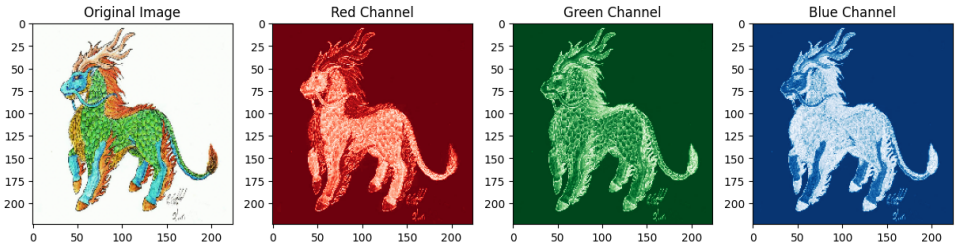


Figure B.1: Viewing the original image and its RGB channels. While the top-left image displays the original image, the other three images show separate channels for red, green, and blue.

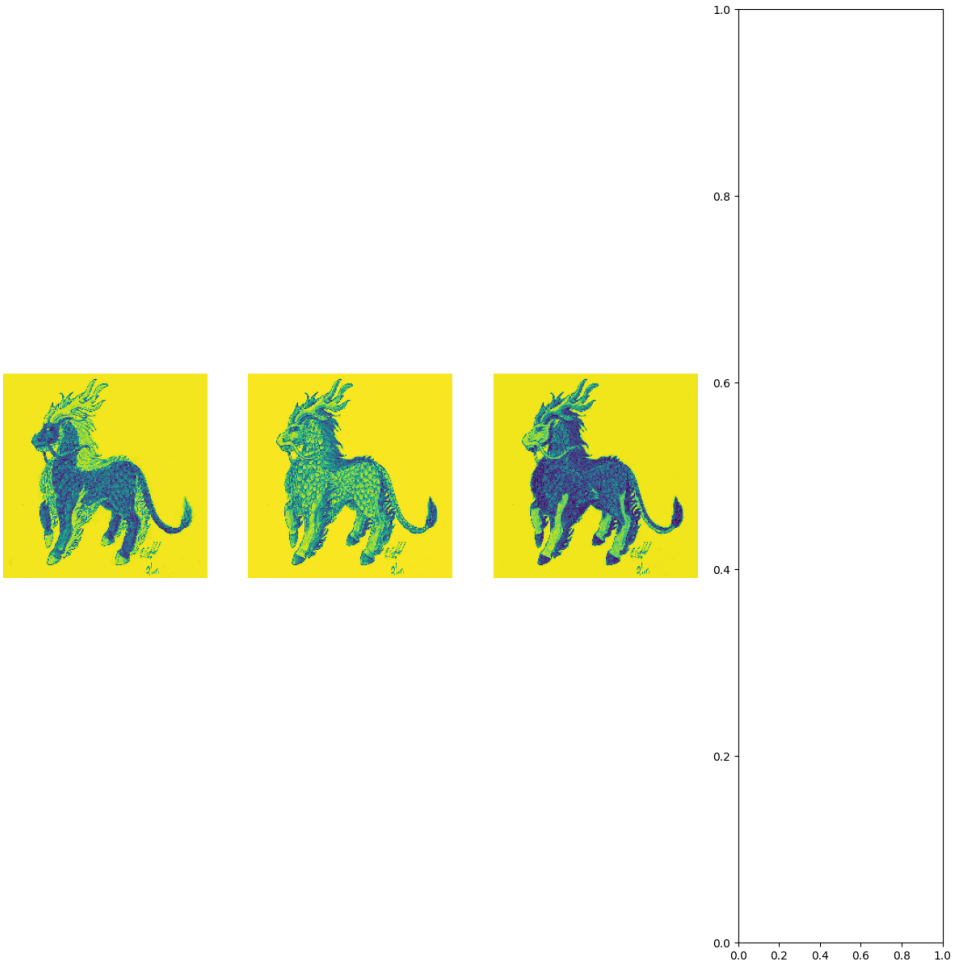


Figure B.2: The activations from the first convolutional layer can be visualised. Each sub-image is a representation of a different filter in the first layer, which highlights various features that the model detects.

Appendix C

User Documentation

In this section, there are detailed instructions on setting up, running, and evaluating the mythological creature classification system, both in Google Colab and on a local GPU machine.

1. Introduction to the System

A modified MobileNetV2 CNN model is used by the mythological creature classification system to categorise five distinct types of mythological creatures: Centaur, Griffin, Minotaur, Pegasus, and Hydra. The project consists of two key notebooks:

- `hyperparameters-mythological-beast-vision.ipynb`: Used to tune hyperparameters for model training.
- `mythological_beast_vision.ipynb`: Contains the core model for training and testing on the mythological creatures dataset.

The trained models are saved in the following directories:

- Hyperparameter-tuned model: `hyperparams-model/`
- Standard model: `models/`

2. System Requirements

- **Operating System:** Windows, Linux, or macOS.
- **Python Version:** Python 3.8 or higher.
- **Libraries/Dependencies:**
  - TensorFlow >= 2.0
  - Keras
  - NumPy
  - Matplotlib
  - OpenCV
  - scikit-learn

- Google Colab (for cloud execution)
- CUDA-enabled GPU (for local GPU execution)

3. Running on Google Colab

Google Colab provides free GPU access, making it a good option for training models on large datasets.

Steps to Run the Code in Google Colab:

1. **Access Google Colab:** Open your web browser and navigate to Google Colab.
2. **Upload the Notebooks:** Upload the notebooks `hyperparameters-mythological-beast-vision.i` and `mythological_beast_vision.ipynb` to Colab by clicking on the "Upload" button under the "File" menu.
3. **Setting up the GPU:**
   - Go to **Runtime** > **Change runtime type**.
   - Set **Hardware accelerator** to **GPU**.
4. **Install Dependencies:**

   ```
   !pip install tensorflow keras opencv-python scikit-learn matplotlib
   ```

5. **Load the Dataset:**
   - Upload your dataset to Colab or link it from a cloud service like Google Drive. You can mount Google Drive using the following code:

     ```
     from google.colab import drive
     drive.mount('/content/drive')
     ```

   - Once mounted, navigate to the location of your dataset.
6. **Running the Hyperparameter Tuning Notebook:**
   - Open `hyperparameters-mythological-beast-vision.ipynb`.
   - Run all cells sequentially to train the model with different hyperparameter configurations.
   - After tuning, the trained model will be saved in the `hyperparams-model/` directory within Colab.
7. **Running the Core Model Notebook:**
   - Open `mythological_beast_vision.ipynb`.
   - Run all cells to initiate the training and save the model in the `models/` directory.

8. **Model Evaluation:**
    - After training, you can evaluate the model's performance using validation data. The notebook will generate confusion matrices and other metrics.
    - You can also visualise the model's predictions using provided sample images.
9. **Download the Model:**

```
from google.colab import files
files.download('/content/drive/mytho-beast-cnn/models/models.h5')
```

4. Running on Local GPU Machine

For users with access to a local machine equipped with a CUDA-enabled GPU, follow these steps:

Steps to Run the Code on Local GPU:

1. **Install CUDA and cuDNN:** Ensure you have installed CUDA and cuDNN compatible with TensorFlow. You can check the official TensorFlow GPU support page for detailed instructions.
2. **Set up a Virtual Environment:**

```
python3 -m venv myth-creature-env
source myth-creature-env/bin/activate  # For Linux/Mac
myth-creature-env\Scripts\activate  # For Windows
```

3. **Install Dependencies:**

```
pip install tensorflow-gpu keras opencv-python scikit-learn matplotlib
```

4. **Clone the Repository or Copy Files:** Copy the notebooks `hyperparameters-mythologicalbeastvision.ipynb` and `mythological_beast_vision.ipynb` to your local machine.
5. **Set Up Dataset:** Ensure the dataset is available on your machine and placed in the correct directory. Modify the paths in the notebooks to point to the dataset on your local system.
6. **Running the Hyperparameter Tuning Notebook:** Open `hyperparameters-mythologicalbeastvision.ipynb` in your Jupyter environment or directly in your IDE (e.g., VSCode, PyCharm). Execute the notebook to train the model with different hyperparameter settings. The tuned model will be saved in the `hyperparams-model/` directory.
7. **Running the Core Model Notebook:** Open `mythological_beast_vision.ipynb` and execute the notebook to train the core model. The final model will be saved in the `models/` directory after training.

8. **Training the Model with GPU:** If TensorFlow detects the GPU, it will use it automatically. You can verify this by running:

```
from tensorflow.python.client import device_lib
print(device_lib.list_local_devices())
```

9. **Model Evaluation:** The notebook includes code to generate evaluation metrics and visualise predictions. Execute the respective cells after training is complete.

5. Contact and Further Assistance

If you encounter any issues or need further assistance, you can contact the project author at [nj585@bath.ac.uk]. For updates or to contribute to this project, please visit the project repository on GitHub.

Appendix D

Raw Results Output

D.1 GitHub Repository Link

https://github.bath.ac.uk/nj585/mytho-beast-cnn

D.2 Validation Result

D.3 Prediction Result

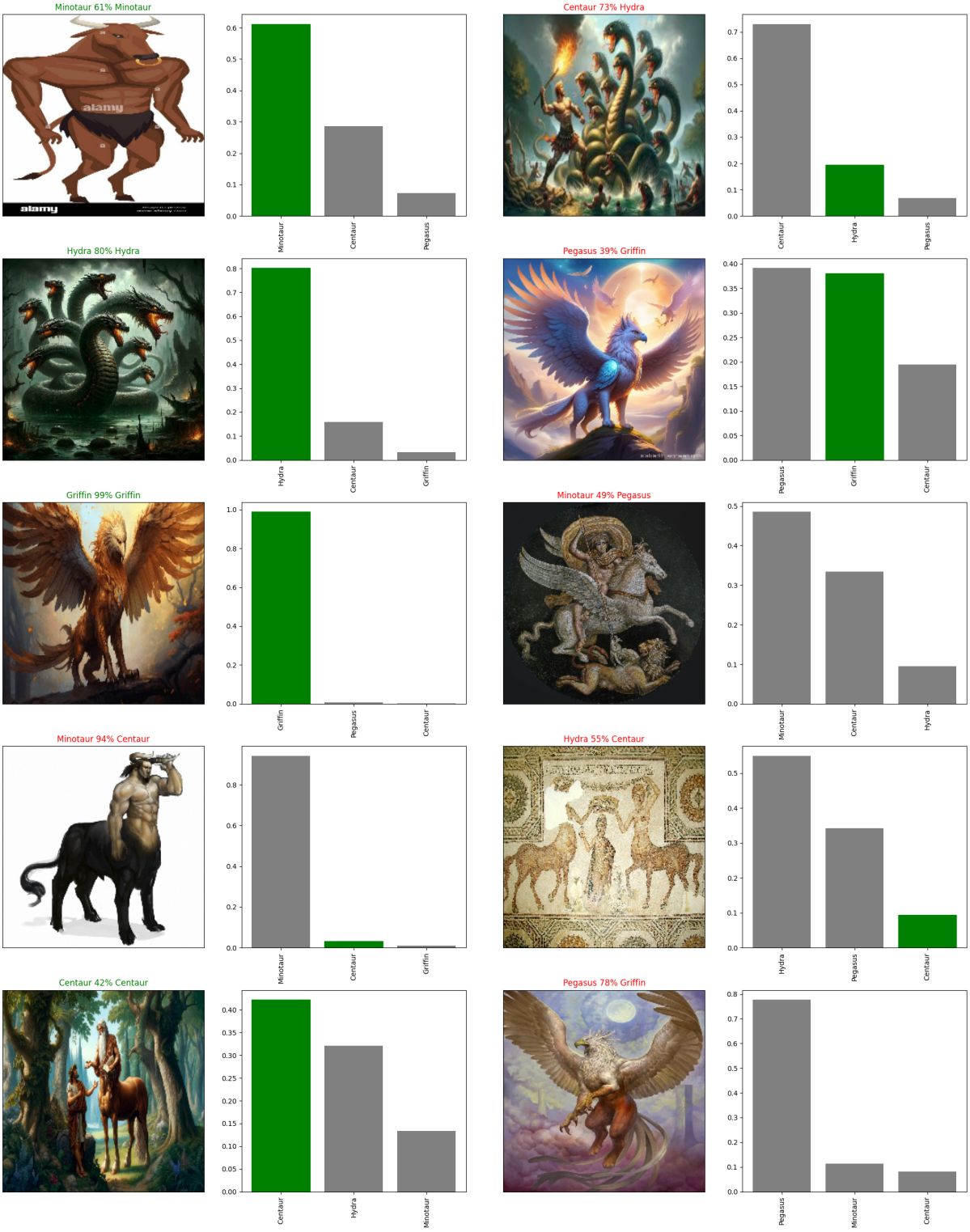


Figure D.1: Image validation prediction output.

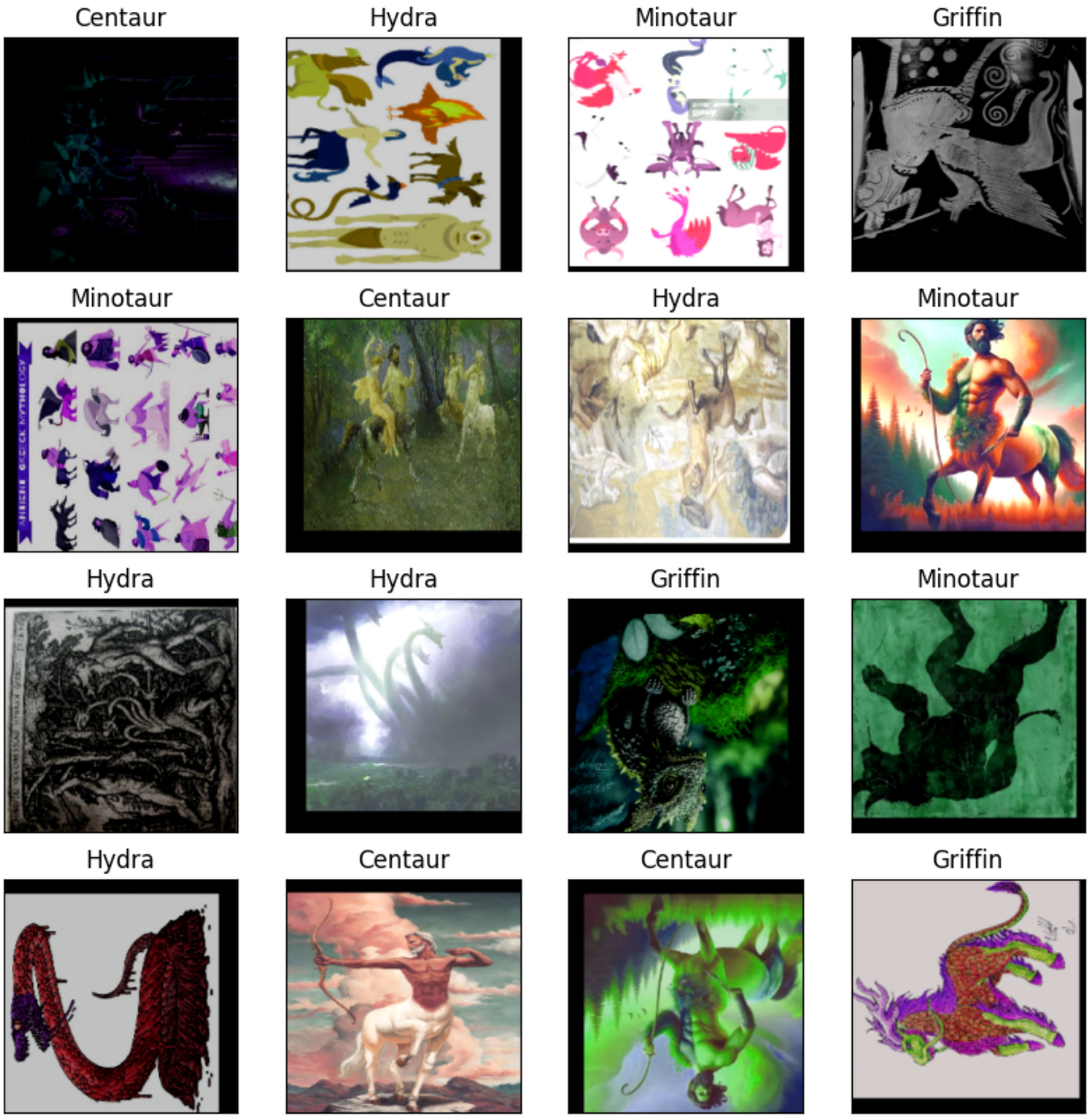


Figure D.2: Image test prediction output.