Pamantasan ng Lungsod ng Pasig
Alkalde Jose St. Kapasigan, Pasig City
College of Computer Studies

Pet Feed Volume Counter Using Convolutional Neural Network for Superworms.PH

A Capstone Project Presented to
Pamantasan ng Lungsod ng Pasig
Division of Pasig City

In Partial Fulfillment
of the requirement for the subject
Software Engineering
S.Y 2023-2024

Prepared by:
Leader: Perin, Mariel B.
Members:
Azarcon, Christian Jull G.
Rances, Francis Luis D.

Doc. Alexen A. Elacio
Professor

June 2024

CHAPTER 1

The Problem and Its Background

A. Introduction

As time progresses, the crucial role of pet feed shops in meeting the diverse dietary needs of animals remains unwavering. Pet feed shops are not only thriving but also evolving with the assistance of advancements in modern technology like using online platforms, as well as the shifting of preferences of consumers. According to Hanjaya, M. et.al. (2019), nowadays, the purchasing behavior from the consumers in the retailing industry has changed and it has been considered as a noteworthy directional change from physical stores buying behavior to web based acquiring behavior. With this transition, the focus on accuracy and efficiency in online shops has become even more noticeable. Every order placed by the owner and delivered to consumers is an important transaction that reflects not only the product quality but also the establishment's dependability and professionalism.

Machine learning continues to show itself as a tool for tackling complex issues and creating innovation in other sectors. Convolutional neural networks, or CNNs, are a machine learning technique that stands out as a remarkable development in the field of artificial intelligence. This machine learning is a class of artificial networks which have emerged as a dominant in numerous computer vision applications that sparks interest across diverse industries. Designed to autonomously and flexibly learn spatial hierarchies of features through backpropagation, CNNs utilize a series of building blocks including convolutional layers, pooling layer, and fully connected layer (Yamashita,R., et. al., 2018). Convolutional neural networks (CNN) are the foundation of computer vision systems that are increasingly being used in animal-related sectors to improve animal management techniques. Still, there's a lot of room for more research and development about the abilities, constraints, and solutions related to these uses. Li, G., Huang, Y., et. al.(2021) has applied the CNN for animal farming.Their study focuses on five important deep learning computer vision tasks: object recognition, posture estimation, semantic/instance segmentation, picture classification, and tracking. Its objective is to do a thorough assessment of CNN-based computer vision systems used in the livestock industry. With the success of their study, they manage to represent significant opportunities for every animal or pet industry to modernize their operations and increase efficiency.

As the pet feed industry undergoes technological advancements, these modern technologies will help to make their operations more organized and efficient. Especially for the online pet feed shop, Superworms.PH, which encounters challenges on managing and achieving their goals to maintain good quality of service to their customers. Manually counting their living products brings hassle and consumes a lot of time. Idea of having a volume counter by leveraging Convolutional Neural Networks and continuously embracing online platforms, they may aim to manage their shop more effectively, accurately, and enhance customer satisfaction. This integration of technology reflects a commitment to excellence and professionalism, guaranteeing that every order placed and delivered reflects the best possible level of quality and service.

B. Background of the Study

Superworms.PH’s journey in the pet feed industry has been marked by commitment to quality and customer service. They started to cater to a diverse range of customers with varying pet needs, from reptiles to birds on February 12, 2021 using a social media platform. Customers demand specific quantities of feed tailored to their pets’ requirements. The owner of the Superworms.PH uses a manual counting process, which becomes their significant challenge in running their business. There are times where they can’t handle multiple orders because of time insufficiency and because of the tiredness they experience in manually counting each customer's order; and there are also times where they experience sending the wrong count of orders to their customers. In short, The labor-intensive task of counting and measuring pet feed not only consumes valuable time but also introduces the possibility of errors, leading to discrepancies in order fulfillment.

Product counting is one of the most important processes in the business industry. Precise product counting is essential to guarantee that the right number of products are packed, protecting against possible consequences like double handling, recounting, delivery delays, and general time waste for business and its customers. As per Ablidas, M. et. al. (2019) one of the primary goals of business transactions is to provide the consumer with the proper number of things. A significant improvement in the packaging system, particularly the product counting process, will be critical in accomplishing this goal. As a result, systematic procedures and techniques for counting products using technology are being pursued.

Because of this, Group 3 of BSCS-2A, students of Pamantasan ng Lungsod ng Pasig S.Y. 2023-2024, came up with a solution that will help the owners of Superworms.PH to be more efficient and accurate. This seeks to transform Superworms.PH's inventory management by automating the counting process and utilizing cutting-edge machine learning methods, especially with the usage of Convolutional Neural Networks. This novel technique guarantees accuracy and dependability in order fulfillment, which not only promises to improve the entire client experience but also expedite operations.

C. Objective of the Study

1. **To develop** a "Pet Feed Volume Counter" system using Convolutional Neural Network (CNN) technology using waterfall development methodology.
2. **To apply** machine learning algorithms, specifically Convolutional Neural Network (CNN) technology.
3. **To integrate** technology stack comprising software frameworks, hardware components such as machine learning models.

4. **To evaluate** the software quality of Pet Feed Volume Counter Using Convolutional Neural Network in terms of:
    a. Functionality Suitability
    b. Usability
    c. Reliability
    d. Performance Efficiency
    e. Security

D. Significance of the Study

This study introduces a unique method to the pet feed industry, using advanced machine learning techniques to precisely count and manage specific orders, thus improving operational efficiency for pet owners and businesses. This study would be beneficial to the following individuals and organizations:

Business Owners. This system minimizes manual effort and time needed for pet feed volume management, allowing Superworms.PH owners to prioritize other business aspects. Through automated counting via CNN technology, it enhances efficiency, reduces errors, and boosts customer satisfaction.

Community (Pet Owners and Customers). This technology's implementation benefits not only Superworms.PH owners but also pet owners and the wider community. It guarantees precise and reliable measurements of pet feed volumes to what they need, improving nutrition management for pets and thereby enhancing their overall well-being.

Student Researcher. This project showcases our expertise in cutting-edge technologies and practical application skills. It not only boosts our academic and technical understanding but also readies us for future endeavors in artificial intelligence and IoT integration.

Future Researchers. The success and effectiveness of this project serve as a valuable reference and inspiration for future researchers exploring similar applications of CNN technology in automated counting and management systems. It contributes to the advancement of machine learning solutions in various industries, showcasing potential for innovation and efficiency improvements in business operations.

E. Scopes and Limitations

In the research titled "Pet Feed Volume Counter Using Convolutional Neural Network for Superworms.PH", the main concepts are the **Pet Feed Volume Counter**, **Convolutional Neural Network (CNN)**, **Superworms**, and **Superworms.PH**. The Pet Feed Volume Counter is the application being developed or studied, which simplifies the process of measuring or counting specific orders and includes inventory sorting for pet feeds. The Convolutional Neural Network (CNN), a type of deep learning model used in image recognition and processing, is the main tool used to develop the pet feed volume counter. Superworms are the specific type of pet feed being studied, and their characteristics could affect how the CNN performs. Lastly, Superworms.PH refers to the context or scope of the study, indicating a specific company. These main concepts guide the research questions and objectives of the study. Moreover, the study seeks to streamline the order and inventory process of pet feeds, specifically superworms, by employing a convolutional neural network for image recognition. This would enable the identification of each superworm in an image. This can be achieved through a software application that the client will utilize upon receiving an order, eliminating the need to count each piece every time an order is placed, thereby accelerating and simplifying business operations. The study is scheduled to commence in the third week of April 2024, marking the start of the final term, at the locations of Superworms.PH and Pamantasan ng Lungsod ng Pasig during the Software Engineering I course.

This research could potentially face several issues and challenges that researchers might face during the study that may influence or impact the results and interpretations of the results. Due to design constraints, this study does not cover several aspects. It does not include superworms in the image that are **overlapping** or touching each other, as this could complicate the image recognition process. **Poor quality images** are also outside the scope of this research, as they could affect the accuracy of the Convolutional Neural Network. **Images without a clear background** are not considered, as they could interfere with the identification of the superworms. **Dead superworms** are also excluded from the study, as their physical characteristics may differ from those of live superworms, potentially affecting the performance of the CNN. Lastly, the study, for the time being, **prioritizes the development of the software application** over other aspects. These limitations are important to note as they define the boundaries of the current research. Some other aspects that could potentially affect the study includes Convolutional Neural Networks (CNNs) requiring high computational resources and large datasets for training to achieve high accuracy rates. CNNs also have difficulty classifying images that are not standardized or contain some degree of tilt or rotation. The accuracy of volume measurements can be affected by human error, equipment limitations, and environmental factors such as temperature and pressure. If these factors are not controlled or accounted for, they could introduce errors into the volume measurements. Training a CNN can be a time-consuming process. If the study is under tight time constraints, this could limit the amount of training that can be done and thus the performance of CNN. Designing and implementing a CNN requires a certain level of expertise in machine learning and computer vision. If the researchers do not possess this expertise, it could limit the effectiveness of CNN. Studying superworms could present challenges due to their biology and ecology. For instance, their behavior, life cycle, and nutritional value could vary under different conditions, which might affect the accuracy of the pet feed volume counter. Additionally, the availability and quality of superworms could also pose a limitation. This is because CNNs typically recognize patterns based on the training data, and overlapping or touching superworms may present a pattern that the CNN has not been trained to recognize. The quality of the image taken can also impact the accuracy of the image recognition process. Factors such as lighting, focus, angle or distance and resolution can affect the ability of the CNN to accurately recognize and measure the superworms.

F. Operational Definition of Terms

The following is the list of terms that will provide meaning to understand the contents of this study:

Artificial Intelligence (AI)- refers to the theory and development of computer systems capable of performing tasks that historically required human intelligence, such as recognizing speech, making decisions, and identifying patterns. It is often used to describe a wide range of technologies that power many of the services and goods we use every day. AI is an umbrella term that encompasses a wide variety of technologies, including machine learning, deep learning, and natural language processing.

Convolutional Neural Network (CNN)- a type of deep learning neural network that is well-suited for image and video analysis. CNNs use a series of convolution and pooling layers to extract features from images and videos, and then use these features to classify or detect objects or scenes

Machine Learning- often abbreviated as ML, is a subset of artificial intelligence (AI) that focuses on the development of computer algorithms that improve automatically through experience and using data. It uses algorithms trained on data sets to create self-learning models that can predict outcomes and classify information without human intervention. These models identify patterns and relationships in data, becoming more accurate as they receive additional data.

Superworms.PH- an online business on Facebook located at pinagbuhatan, pasig city where they sell superworms, a feeder for animals like fish, hamsters, and tarantulas.

Volume Counter- a device or system that is used to count the number of items that are in a certain process or area, which is very important in the management of stocks, quality assurance, and monitoring of production.

Convolutional Layers- A convolutional layer is the main building block of a CNN. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training.

Pooling Layers- also known as downsample layers, are an essential component of convolutional neural networks (CNNs) used in deep learning. They are responsible for reducing the spatial dimensions of the input data, in terms of width and height, while retaining the most important information.

Segmentation- a computer vision technique that partitions a digital image into discrete groups of pixels—image segments—to inform object detection and related tasks

Image Recognition- a sub-category of computer vision technology that deals with recognizing patterns and regularities in the image data, and later classifying them into categories by interpreting image pixel patterns.

CHAPTER 2

REVIEW OF RELATED LITERATURES AND STUDIES

This chapter covers the literature that is pertinent to the current study's premise. It also examines several studies and resources from the Internet to offer the necessary background and knowledge for the completion of this research. To better explain this study, this chapter gathered claims from previous studies and works related to the current study under the following key points: (1) machine learning, (2) object detection and counting, (3) datasets, (4) single-object counting vs. multiple-object counting, and (5) image recognition.

Development Methodology

The Waterfall model, also called a straight-line life cycle model, shows how to make software step by step in a straight path. In this way, one stage must end before the next one starts, and stages don't mix. This approach is often used in making software to help ensure a project goes well by splitting the work into clear, separate steps, each step setting the stage for the one after it.These steps start with Requirement Analysis, where we list out what the software should and shouldn’t do clearly; then move to Design, where we lay out plans for how the software will solve problems, including detailed ideas for algorithms, system structure, data keeping, and interfaces; next is Implementation, where we turn those plans and needs into real, working software through coding and setting it up; followed by Testing, where we check to make sure the software does what we wanted it to do and works right by finding and fixing errors; and lastly, Operation and Maintenance, which is about making changes after the software is out to make it run better, solve any problems, and change it as users' needs or the surroundings change(Senarath et al. (2021) and Adenowo (2020))

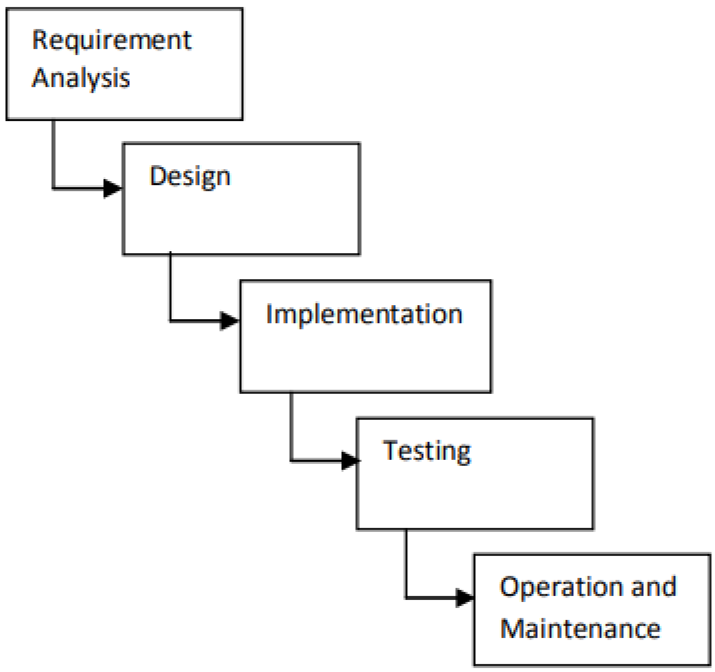


Figure 1 Waterfall Software Development Methodology.

One such structured and systematic approach in software development is the Waterfall approach, as described in the work of Priyanka Sharma et al. (2021), which is particularly helpful when developing software that contains algorithms such as CNNs. This method focuses on the linear design approach which is appropriate when applying CNN based systems in tasks such as object identification and categorization. The Waterfall model also has phases that are as follows: Requirement Analysis, Design, Implementation, Testing, Operation and Maintenance; the model ensures that each phase is fully completed before moving to the next phase. This minimizes the probability of missing out on important aspects and also guarantees a thorough understanding of the project specifications. In particular, in the projects that involve CNN, this approach helps to ensure that requirements are gathered and analyzed carefully, the architecture is designed, the algorithm is implemented, and the model is tested to meet the initial specifications as closely as possible. The Waterfall model is especially beneficial in academic environments where it is necessary to adhere to clear and documented processes and where the model's linear structure is a strength, as it was in Sharma's study on predicting heart disease with machine learning methods (Sharma et al. , 2021). This in turn provides a solid foundation for building CNN-based applications that are dependable and precise.

CNN is a well-known trained network that enhances its classification efficacy by using datasets from open-source networks. It was trained on millions of images from the CIFAR-100 and Image-Nets datasets. The datasets that are utilized consist of millions of little photos. They can therefore effectively and accurately simplify, which enables them to classify the out-of-sample samples for the classes. [1]

The CIFAR-100 dataset consists of a wide range of general object images organized into numerous super-classes, each containing several subclass categories. This dataset includes a total of 100 classes, with each class comprising 600 images [2].For the proposed project, specific categories from the superclasses of Household furniture and vehicle are selected for training the networks. Additionally, the ImageNet dataset, organized according to the WordNet hierarchy with meaningful concepts, is used. It contains super-classes of images further divided into subclasses.[1]

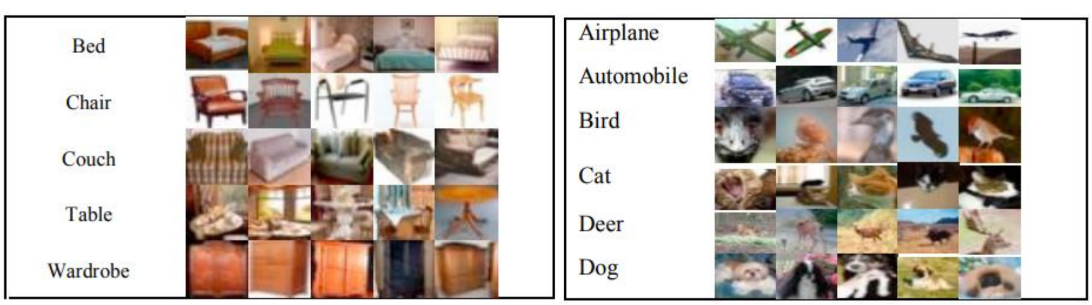


Fig 2 : Few classes of CIFAR10 and CIFAR100 Datasets

The dataset CIFAR-10 contains 60000 photos total—32x32 color images—were grouped into 10 classes, with 6000 images in each class. The collection of data includes 10,000 test photographs and 50,000 training images. Five training batches and one test batch make up the dataset. each containing 10,000 pictures. From each class, a random selection of test photographs is made.[1]

Table 1. Performance of CNN's on CIFAR100 test dataset

CIFAR-100		AlexNet	GoogLeNet	ResNet50
Image Category	Bed	0.00%	70.80%	49.60%
	Bicycle	21.0	74.2%	55.00%
	Bus	84.00%	63.20%	36.80%
	Chair	90.00%	89.60%	57.60%
	Couch	11.00%	14.60%	76.40%
	Motorcycle	95.00%	74.60%	99.20%
	Streetcar	21.00%	0.84%	63.80%
	Table	00.00%	73.60%	33.40%
	Train	30.00%	95.60%	34.20%
	Wardrobe	89.00%	89.40%	92.20%

Table 2. Performance of CNN's on the CIFAR10 test dataset

CIFAR-10		AlexNet	GoogLeNet	ResNet50
Image Category	Airplane	41.80%	51.10%	90.80%
	Automobile	21.80%	62.10%	69.10%
	Bird	00.02%	56.70%	72.60%
	Cat	00.03%	78.80%	61.90%
	Deer	87.60%	49.50%	75.40%
	Dog	23.00%	57.50%	82.10%
	Frog	24.20%	90.20%	76.60%
	Horse	34.70%	78.20%	84.70%
	Ship	31.70%	95.50%	83.20%
	Truck	95.90%	97.10%	84.60%

Deep learning is facilitated by convolutional neural networks algorithms that combine input photos with kernels or filters to extract characteristics. An NxN picture is convolved using this convolution procedure and a fXf filter learns the same feature across the board in the image [4]. As the window slides upon each action, the characteristics are discovered by the feature maps. The feature maps depict the local picture receptive field and use shared [3].

Table 3: The most common and successful CNN architectures for image classification.

Model	Trainable layers	Main specifications
AlexNet	8	5 convolutional layers and 3 fully-connected layers. [5]

VGG-16	16	13 convolutional layers with 3x3 filters, and 3 fully-connected layers. [5]
GoogLeNet	22	created a module for initiation that significantly lowers the quantity of parameters while attaining elevated precision. On top, average pooling is employed. rather than being totally connected, of CNN strata. [5]
ResNet-50	50	A framework for deep residual learning, skip both batch normalization and connectivity. Far more profound than VGG-16 (50 as opposed to 16) but possessing lesser complexity as well as improved output. [5]

Convolutional Neural Network

In the contemporary period, there is fierce competition among businesses throughout the world. Enterprises are increasingly turning to big data and predictive analytics to develop efficient profit-boosting strategies, and the use of such technology is expected to grow in the future. As a result, developing efficient supply chain strategies is a critical aspect in streamlining corporate operations, increasing operational efficiency, improving customer relationship management, and enabling market sustainability. According to Joseph, Reuben Varghese, et al.'s (2022) study, "A hybrid deep learning framework with CNN and Bi-directional LSTM for store item demand forecasting," there has been a significant increase in the use of CNN-LSTM-based architecture when dealing with numerous sequence prediction problems (such as videos or images). Although demand forecasting is the architecture's most well-known use, it is also used in many other sectors. Because it combines the sequence prediction of LSTM with the feature extraction capabilities of CNN to increase demand forecasting accuracy, the CNN-LSTM-based architecture is incredibly robust and efficient. Furthermore, there are other uses for this design.

Recent advances in deep learning, notably Convolutional Neural Network (CNN) algorithms, have outperformed traditional object identification systems in accuracy. However, their resource-intensive nature presents difficulties for mobile deployment because of restricted processing power and memory. Yanai K. et al. (2016) address this by looking at CNN architectures designed for mobile devices and presenting a multi-scale network-in-networks (NIN) method that allows users to balance recognition accuracy and speed. They employ specific libraries such as the BLAS library for iOS and the NEON SIMD instructions for Android to improve convolutional layers while designing mobile apps for iOS and Android, respectively. Their detailed research covers memory and computational limits, emphasizing the significance of platform-specific adaptations and picture preprocessing approaches for effective CNN implementation in mobile apps. The suggested multi-scale NIN architecture, when combined with better convolutional layer processing, has the potential to successfully handle mobile object identification challenges.

Similar System

Object detection is so close to video analysis and image understanding that it has attracted much research attention in recent years. Traditional object detection techniques rely on handmade features and shallow trainable structures. With the rapid development in deep learning, more powerful tools, which are able to learn semantic, high-level, deeper features, are introduced to address the problems existing in traditional architectures (ZQ. Zhao et al., 2019). Which makes convolutional neural networks to be part of these representative tools.

Since cameras are the main source of visual data utilized by computer vision systems, they are essential to object detection. In the domain of deep learning-based object detection, two main approaches have emerged: anchor-based and anchor-free methods. Anchor-based methods include one-stage and two-stage detection algorithms, where one-stage methods predict object properties directly from predefined anchor boxes, while two-stage methods generate region proposals before refining object boundaries. In contrast, anchor-free methods, like Keypoint-based detection (e.g., FCOS), rely on detecting key points to define bounding boxes. Generally, one-stage methods prioritize speed, while two-stage methods offer higher accuracy. In the study of M. Tan et al.(2022) entitled "Animal detection and classification from camera trap images using different mainstream object detection architectures", they employed YOLOv5, FCOS, and Cascade R-CNN to detect animals in a complex forest environment, with experiments conducted jointly and separately on day and night images to evaluate model performance in different lighting conditions.

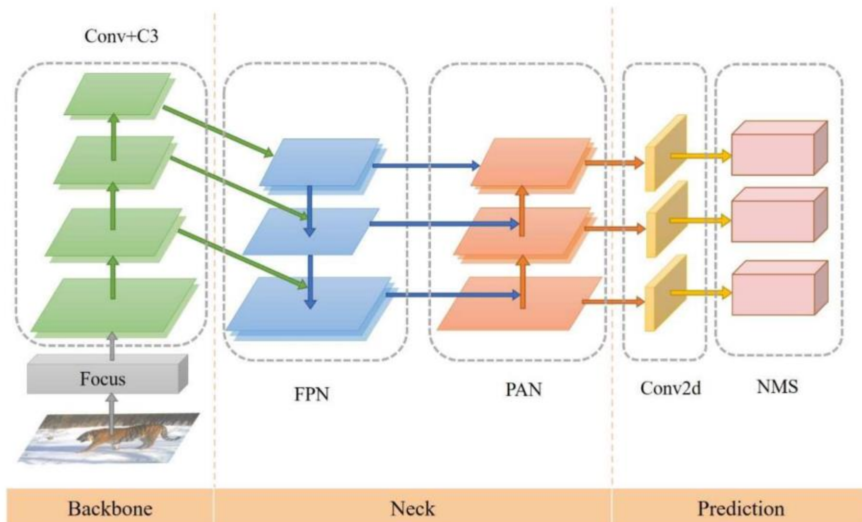


Figure 3. YOLOv5 structure diagram. Conv is convolution; C3 is improved from the Cross Stage Partial Network (CSP Net); Conv2d is two-dimensional convolution.

The Internet of Things (IoT) and Artificial Intelligence (AI) have been employed in a variety of sophisticated system applications, including transportation, robotics, industrial, and automation. According to Ramalingam, B., et al. (2021), the IoRT (Internet of Robotic Things) is a more advanced version of the Internet of Things and has been connected to deep learning-based items such as counters. The deep learning-based object counting problem aims to determine the number of object occurrences in a single picture or video sequence. It has several real-world uses. Balakrishnan demonstrated the Faster RCNN ResNet101 method, which was utilized to determine the number of false ceilings.

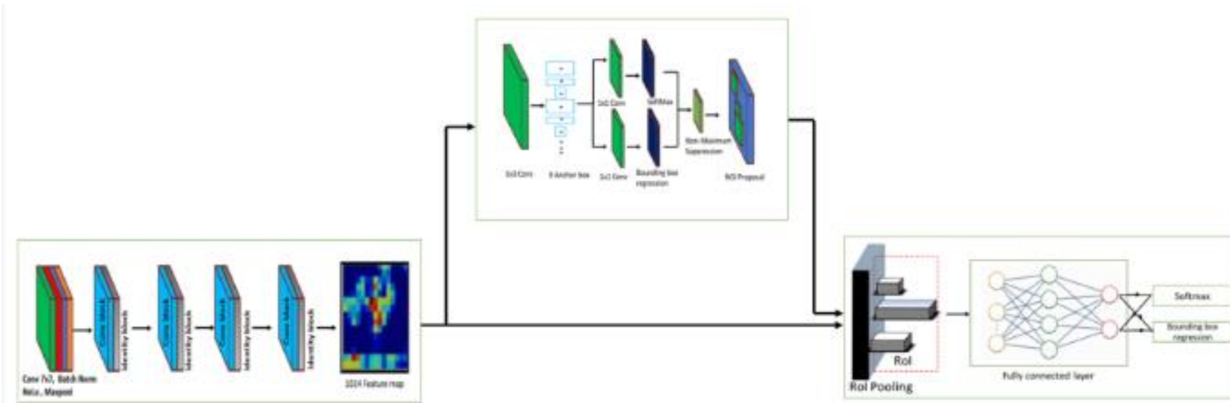


Figure 4. Faster R-CNN RestNet architecture.

ResNet 101 deconstructs pictures to better comprehend them, generating a feature map for object detection. The RPN proposes object positions based on predefined boxes, changing and improving estimations. The Detector and Classifier Head serve as the brain, verifying items and their positions while eliminating assumptions for a clear image.

In the field of computer vision and artificial intelligence the task of object counting stands as a fundamental challenge with a variety of real-world applications. The distinction between single-object counting and multiple-object counting becomes obvious as researchers continue to explore this area. Both seek to quantify objects in an image, but the differences are in how difficult the task is. Single-object counting is the exact counting of each individual object in an image, while multiple-object counting is the simultaneous counting of different instances of different objects. Understanding this difference is important in designing effective algorithms particular to specific applications ranging from tracking animals to inventory management and beyond. This study not only explains the complexities of object counting but also opens the door for creative solutions to deal with a range of problems in various industries.

Although the field of object counting is well-established, Huberman-Spiegelglas et al. (2022) focus on a particular subset of it: single-object counting in single images. Their method addresses situations where objects repeat within an image in contrast to traditional object counting which frequently involves numerous different types of objects and possibly large quantities. The study's active learning approach to weakly-supervised learning is its strongest point. This allows the network to achieve good results with minimal labeled data, a significant advantage for single-object counting where extensive annotation can be time-consuming. Also, by examining the internal representations of the network, they are able to pinpoint informative data points during active learning which may result in counting that is both quicker and more precise. However, it's important to consider limitations. The approach might not apply well to situations with different objects, different quantities, or overlapping objects, which are frequent difficulties in more comprehensive object counting tasks. It is intended for single images with repeating objects. Also, though active learning has advantages, it requires more steps than traditional methods, which could make it less useful for real-time applications.

Seguí et al. explore the use of Convolutional Neural Networks (CNNs) in object counting, with an emphasis on differentiating between counting a single object and counting multiple objects. They propose a novel approach where CNNs learn object representations indirectly through a "learning to count" strategy. Experiments on counting even handwritten numbers and pedestrians in photos are used to illustrate this concept. According to their research, concentrating on single item counting for example, counting even digits can result in generalized object representations that are advantageous for more comprehensive recognition tasks. The authors create a CNN architecture that consists of two stages, the first stage counts item occurrences precisely, while the second stage agnostically captures underlying object attributes. It highlights CNNs potential for object counting and opens the door for more research in this field. The approach shows effectiveness in both synthetic and real world scenarios.

Uchida (2013) noted in his study that image processing and pattern recognition techniques are important in analyzing bioimages, offering numerous options for researchers. A general purpose of image pattern recognition is to classify an image or a target object or a region into one of types, i.e. classes. While achieving human-level recognition accuracy is challenging, image pattern recognition has been applied to a number of different applications already. Optical character recognition (OCR) is one of the most classic applications, where an image of a single character is classified into one of 52 classes. Recognition of faces and more general visual objects is also a very active recent research topic. Choosing the best technique for a given task becomes crucial due to the wide range of techniques available. For instance, different binarization techniques have different properties, so choosing the best method for a particular application requires careful thought. Research in image processing and pattern recognition is moving forward steadily, offering gains in computational efficiency, accuracy, robustness, versatility, and usability much like advances in biology. For many biological tasks, fully automated image analysis may be possible with future techniques. These methods can also reveal new discoveries and provide empirical support for established biological phenomena. Continued progress hinges on fostering mutual collaboration between biologists and image processing specialists. Such collaboration not only facilitates the improvement of existing techniques but also paves the way for the development of innovative solutions to address emerging challenges in image recognition.

Deep learning is a multilayer neural network learning algorithm that has emerged in recent years, which indicates the beginning of a new era in machine learning and greatly improving artificial intelligence and human-computer interaction. Meiyin Wu and Li Chen (2015) conducted research on image recognition using deep learning techniques that delves into the application of deep learning methodologies to handwritten character recognition, specifically exploring two prevalent algorithms which are Convolutional

Neural Networks (CNN) and Deep Belief Networks (DBN). Through comprehensive performance evaluations on both the MNIST database and a real-world handwritten character database, they assess the efficacy of CNN and DBN. Results indicate that CNN achieved a classification accuracy rate of 99.28% on the MNIST database and 92.91% on the real-world database, while DBN yielded rates of 98.12% and 91.66%, respectively. Their findings underscore the remarkable feature learning capability inherent in deep learning models removing the need for manual feature extraction. Deep learning algorithms exhibit a propensity for capturing natural data features, thereby displaying their effectiveness in handwritten character recognition tasks.

Liu (2018) explores Convolutional Neural Networks' (CNNs') inner workings for image recognition applications. Through his work, he shows the benefits of local connections and weight sharing for efficient feature extraction from images in the CNN recognition process. Additionally, pooling layers are mentioned for enhancing the network's robustness against spatial variations. However, a noted drawback is the difficulty in interpreting the abstract features learned by CNNs due to their "black box" nature. Despite this, the study emphasizes CNNs' remarkable performance in image recognition tasks. Looking forward, the challenge lies in creating more stable computing environments and increasing processing speeds, achievable through advancements in both hardware and software. While fully understanding the inner workings of these networks remains a hurdle, CNNs undeniably offer exceptional capabilities for image recognition.

Software Quality Assurance

From the study conducted by Binta Islam, S., et al. (2023) entitled “Animal Species Recognition with Deep Convolutional Neural Networks from Ecological Camera Trap Images,” software quality assurance (SQA) is crucial in developing and applying species recognition. The paper also discusses the data augmentation methods that can be used to improve the generalization of the models and avoid overfitting. To ensure model reliability, cross-validation and performance measures such as accuracy and F1-score are used. Moreover, the SQA practices include the automated testing, the version control, the documentation, and the continuous performance.

The article “Using Convolutional Neural Networks for Image Recognition,” by Hijazi, S. , Kumar, R. & Rowen, C. (2015) discusses the utilization of CNNs for image recognition and outlines several important features that are pertinent to software quality assurance. It highlights the need to preprocess the data to improve the quality of data used in building models to improve on the robustness of the models. Cross validation and other performance metrics are important in ensuring that the CNN models developed are reliable and effective. Proper error detection and management, model updates, versioning, and documentation are critical to address problems and guarantee replicability. Ongoing performance monitoring is emphasized as critical to detecting performance over time, which is in line with SQA principles to improve system dependability and user satisfaction.

CHAPTER 3

METHODOLOGY

Introduction

This chapter presents the method used in identifying and counting superworms using Convolutional Neural Networks (CNNs). This describes the overall strategy, the specific procedures followed, conditions, and the layout of the system. The methodology is designed to make the system precise, fast, and stable in real-world conditions. The main focus of this is on the building of a system with the application of CNN, leveraging its ability to learn and recognize intricate patterns in high-resolution images of the superworms. The procedures include the specification of hardware and software requirements, data gathering processes, and the design of the CNN model. To validate the system’s performance and reliability, verification and testing is discussed here.

H. General Method Used

CNNs are a popular choice for image recognition tasks due to their ability to learn hierarchical features from raw pixel data. They consist of convolutional layers that automatically learn relevant features from input images. CNNs excel at capturing spatial hierarchies and local patterns in images. For superworm detection, CNNs can learn to recognize distinctive features (such as body shape, color, and texture) that differentiate superworms from other objects. Using CNNs aligns with the topic because they provide an effective way to analyze superworm images and count their occurrences. The convolutional layer detects patterns within subregions of the input image using receptive fields (also known as filters or kernels). It slides the receptive field across the input volume, performing element-wise multiplications and summing up the results to produce feature maps. Convolutional layers learn local features (such as edges, textures, and shapes) from the input data. Convolutional layers can have multiple filters, each capturing different features. In superworm detection, a convolutional layer might learn to recognize specific superworm features (e.g., body segments, color patterns).Pooling layers, also known as subsampling layers, systematically reduce the spatial size (width and height) of feature maps produced by convolutional layers. It is used because pooling layers reduce spatial dimensions, making the feature maps more compact. Also Pooling layers enhance robustness to small translations of the input (e.g., slight shifts in superworm positions). Downsampling via pooling layers ensures efficient representation of superworm features. Pooling layers help the model focus on essential features while ignoring minor variations, aligning with accurate superworm counting.

I. Procedure

1. Requirements Specification

A requirements specification makes sure that everyone working on the project knows exactly what needs to be done and can collaborate effectively to achieve the same objectives. It acts as a point of reference throughout the duration of the development process to guarantee that the finished product complies with the specifications and meets the expectations of the stakeholders.

Table 4:

Hardware Requirements

Hardware Requirements:	Justification
Camera: High-resolution camera (1080p or higher) for capturing detailed images of superworms.	The camera used To take detailed pictures of superworms, a high-resolution camera is required, ideally 1080p or greater. The Convolutional Neural Network (CNN) model

	needs enough detail in the images it captures in order to accurately analyze and classify them, and high resolution makes this possible.
GPU: A powerful GPU (e.g., NVIDIA RTX series) for training and running the CNN model efficiently.	The GPU, or graphics processing unit, is necessary for training and operating the CNN model effectively. GPUs are extremely parallel processors that can perform demanding calculations much more quickly than standard CPUs when training deep learning models, such as CNNs. Because of their powerful computing capacity and dedicated tensor cores, the NVIDIA RTX series GPUs are especially well-suited for deep learning applications.
CPU: Multi-core processor (e.g., Intel i7 or AMD Ryzen 7) to handle data preprocessing and other tasks.	The CPU (Central Processing Unit) is Preprocessing data and other operations related to training and operating the CNN model require a multi-core processor. For operations like loading data, preprocessing, and workflow coordination, the CPU is still required, even if the GPU bears the brunt of the severe computational load during training.

Memory: At least 16GB RAM for smooth operation during training and inference.	To ensure seamless functioning during the training and inference phases, a minimum of 16GB of RAM is required. Enough RAM guarantees that the system won't encounter any performance snags or slowdowns when handling the massive amounts of data that are frequently involved in deep learning activities.
Storage: SSD with sufficient capacity (minimum 500GB) for storing images, datasets, and models.	The Storage (SSD) save pictures, datasets, and models, an SSD (Solid State Drive) with enough space is required, ideally at least 500GB. Compared to conventional HDDs (Hard Disk Drives), SSDs have substantially quicker read and write speeds, which is essential for effective data access during training and inference. Furthermore, the huge storage capacity guarantees that the vast datasets and models created during the deep learning process have enough room to be stored.

Table III shows the specification of the hardware needed for the system. The requirements meet the need of acquiring high resolution images, processing, and storage for the CNN to detect and count superworms.

The first of the hardware requirements is the need to have a high-quality camera, with a minimum of 1080p. This camera provides clear images of superworms that are important for the CNN model to analyze and classify them correctly. The reason for this requirement is in the need for high-resolution imagery to allow the CNN to learn and distinguish individual features of superworms for accurate detection and counting.

The second hardware requirement is the need to use a powerful GPU, preferably the NVIDIA RTX series to train and run the CNN model. The parallel structure of the GPU enables the quick computation of calculations needed in deep learning models, thus shortening the training period. The rationale for this requirement is that GPUs, especially those with tensor cores, are essential for performing the computationally intensive operations that are required in CNN training and inference.

The third hardware requirement is a multi-core processor like Intel i7 or AMD Ryzen 7 for data preprocessing and other supportive tasks. The GPU is responsible for most of the computations, but the CPU is required to control the processes, input data, and perform preliminary data processing. The rationale for this requirement is to ensure that these tasks are effectively dealt with in order to enhance the efficiency and continuity of the process.

The fourth hardware requirement is the minimum of 16GB RAM necessary for proper functioning of the model during the training and inference. Sufficient RAM is important to avoid issues such as slow processing or lagging which may be experienced when dealing with big data common in deep learning applications. The rationale for this condition is to keep the system fast and effective in its operations to accommodate large data processing functions.

The fifth and final hardware requirement is an SSD with a minimum of 500GB to store images, datasets, and models. SSDs have much higher read and write speeds compared to the conventional HDDs which is crucial for the proper data access during training and evaluation. The rationale for this requirement is the ability to access and store data as quickly as possible, which enhances the productivity of the workflow.

Table 5:

Software Requirements

Software Requirements	
• **Operating System**: Linux (Ubuntu preferred) or Windows.	

• **Deep Learning Framework**: ◦ **TensorFlow** or **PyTorch** for building and training the CNN model.	
• **Image Processing Libraries:** ◦ **OpenCV** for image preprocessing tasks. ◦ **PIL** (Python Imaging Library) for image manipulation.	
• **Development Environment:** ◦ **Python** as the primary programming language. ◦ **Jupyter Notebook or PyCharm** for coding and experimentation.	
• **Additional Libraries:** ◦ **NumPy** for numerical operations.	

Operating systems such as Windows or Linux variants like Ubuntu are recommended for deep learning workloads. While both Linux and Windows offer strong support for deep learning frameworks and libraries, many deep learning practitioners choose Linux distributions—especially Ubuntu because of their dependability, adaptability, and compatibility with a large number of tools and libraries used in the field.

The Deep learning frameworks are software libraries that offer tools and abstractions for the construction and training of deep neural networks. PyTorch and TensorFlow are two of the most widely used deep learning frameworks. Low-level APIs for fine-grained control over model architecture and training procedure are provided by these frameworks in addition to high-level APIs for designing and training neural networks. Deep learning models are frequently developed using both TensorFlow and PyTorch, and the decision between the two is frequently influenced by a variety of factors, including project requirements, experience level, and personal preferences.

The preprocessing and manipulation of images prior to feeding them into the deep learning model is made possible by the image processing packages. There are two widely used libraries for image processing:

OpenCV, or the Open-Source Computer Vision Library, is a well-liked open-source library with many features and methods for object detection, image processing, feature extraction, and other activities. It is extremely effective and frequently utilized for a variety of computer vision applications in both industry and research.

The Python Imaging Library, or PIL, is a library that allows you to access, work with, and save a wide variety of image file formats. It offers simple picture processing features including cropping, rotating, and resizing. Although PIL is portable and user-friendly, it could not have all the sophisticated features.

CNN Model Requirements

a. **Model Architecture:** A well-suited architecture for object detection and counting, such as:

   i. YOLO (You Only Look Once)

   ii. SSD (Single Shot Multibox Detector)

   iii. Faster R-CNN

b. **Pretrained Models**: Using pretrained models (e.g., on COCO dataset) as a starting point for transfer learning.

c. **Training Data**: A large, annotated dataset of superworm images for training the model.

Table 6:

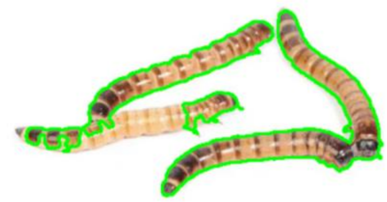
Superworms Image Requirements Specification

Description	**Sample Image**
IMAGE QUALITY **Resolution:** High-resolution images (at least 1080p) to ensure clear visibility of superworms. **Focus:** Sharp focus to distinguish individual worms, especially in crowded samples. **Lighting:** Consistent, even lighting to minimize shadows and glare.	

<table>
<tr><td>IMAGE COMPOSITION

Background: Plain, contrasting background to enhance worm visibility.

Coverage: Ensure the entire frame is filled with worms without significant empty space.

Orientation: Capture images from a top-down perspective for uniformity.</td><td></td></tr>
</table>

Analysis of Non-Functional Needs

Non-functional requirements analysis is a needs analysis outside the function of the system, this analysis consists of hardware analysis, software analysis, and user analysis.

Analysis of Functional Needs

Functional requirements analysis is the process of describing the activities needed by a system so that the system built runs well and in accordance with needs. System modeling is modeled using a use case diagram.

Table 7

Functional Requirements	**Non-Functional Requirements**
Input Handling: The system must accept high-resolution images or video streams. Preprocess images to standardize size,	**Suitability**: The system needs to work well in different places where superworms breed, even if the lighting and surroundings are different. It should be able to handle images from

resolution, and lighting conditions.	different kinds of cameras.
Counting Mechanism: Utilize a trained CNN model to detect and count superworms in real-time. Handle overlapping and varying worm orientations.	**Reliability**: The system needs to work well even when things go wrong, like unexpected inputs or interruptions. It should be able to spot and fix errors while counting.
	Performance Efficiency: The system needs to be fast and efficient, handling many images or video frames quickly. It should use hardware effectively to avoid slowing down or overheating.
Output: Display the count on a user-friendly interface. Allow export of count data in formats like CSV and JSON.	**Usability**: Provide an intuitive interface for non-technical users.
	Security: Ensure data privacy and secure handling of input and output data.

Functional Requirements

The requirement for the system to be able to handle input in the form of either high-resolution photos or video streams is stated in the input handling section. Standardize size, resolution, and lighting. In order to provide correct analysis, the system must preprocess the incoming photos to guarantee uniformity in size, resolution, and lighting.

Counting Mechanism by using a taught Convolutional Neural Network (CNN) model, the system must be able to simultaneously identify and count superworms from the input images or video streams. This criterion specifies that the CNN model must be trained. Accurately count superworms by handling overlapping and different worm orientations: The counting mechanism must be able to handle obstacles like overlapping worms and different orientations.

The final output criterion is that the system must give a user-friendly interface that shows the number of superworms that have been found. Permit users to export count data in common forms like CSV and JSON: In order to facilitate additional analysis or system integration, users should be able to export count data in common formats like CSV (Comma-Separated Values).

Non-Functional Requirements

Suitability is the ability of the system to adjust to diverse lighting and background circumstances as well as different places where superworms are common. It should be able to process photos from various cameras or other devices that are frequently utilized in agricultural or scientific contexts.

Reliability means the system should be able to withstand errors and be strong enough to handle erroneous results or unexpected inputs without crashing. To detect and fix mistakes made during the counting process, it ought to include integrated error detection and recovery techniques.

Performance Efficiency is the ability of the system to process many pictures in an acceptable amount of time. It should be designed with speed and resource efficiency in mind. To guarantee seamless functioning and avoid overheating or performance deterioration, it should make effective use of the hardware resources available, limiting CPU and GPU utilization.

Usability means there should be clear instructions on how to operate the system and an intuitive, user-friendly user interface. While counting, it should give the user feedback on how things are going and let them know if there are any possible problems. Users should be able to use the system from a variety of devices, including smartphones, tablets, and desktop computers, by supporting multi-platform accessibility.

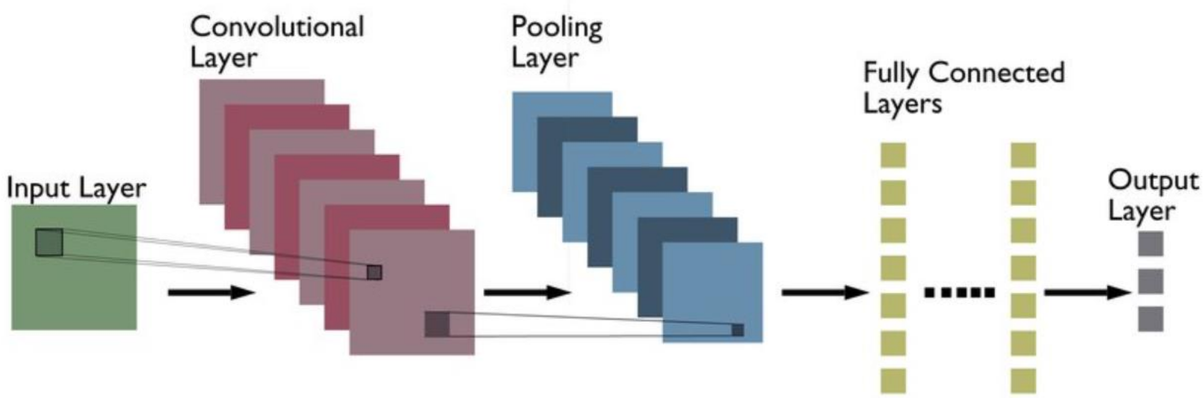
Robust permission and authentication procedures should be implemented by the system to manage access to sensitive information and features. Communication lines should be encrypted to guard against illegal data interception and manipulation while it's being transmitted.

2. Design

Convolutional Neural Network (CNN)

A deep learning algorithm primarily used for analyzing visual imagery. It's characterized by its ability to automatically and adaptively learn spatial hierarchies of features from raw data. In the case study "Pet Feed Volume Counter Using Convolutional Neural Network," CNNs can be employed to analyze images of pet feeding stations. By processing these visual inputs, the CNN can identify and quantify the amount of pet feed on a specific inventory or order, effectively serving as a volume counter. This application leverages the CNN's capability to detect patterns and features within images, enabling it to estimate the volume of pet feeds, specifically superworms based on visual cues. Through training on labeled data, the CNN can learn to associate visual characteristics with specific volume levels, allowing for accurate estimation of pet feeds without the need for manual measurement.

Figure 5: Convolutional Neural Network (CNN) Architecture for Estimating Pet Feed Volume



Convolutional Layer: This layer is responsible for extracting features from the input images. It applies a set of learnable filters (kernels) to the input image, convolving them to produce feature maps. Each filter detects different features such as edges, textures, or shapes.

Activation Layer: Following the convolutional layer, an activation function (such as ReLU, Rectified Linear Unit) is applied element-wise to the feature maps. This introduces non-linearity to the model, enabling it to learn complex patterns and relationships in the data.

Pooling Layer: The pooling layer reduces the spatial dimensions of the feature maps while retaining the most important information. It does this by downsampling the feature maps, typically through operations like max pooling or average pooling, which take the maximum or average value within small regions of the feature maps.

Fully Connected Layer: In this layer, the feature maps are flattened into a vector and connected to a traditional neural network structure. This allows the network to learn high-level features and make predictions based on the features extracted from the previous layers. The fully connected layer is typically followed by an output layer, which produces the final predictions of the CNN.

Python

Being a versatile and widely-used programming language, serves as the foundation for implementing the "Pet Feed Volume Counter Using Convolutional Neural Network" case study. Its simplicity and extensive libraries make it an ideal choice for tasks ranging from data preprocessing to model evaluation and deployment. In this case study, Python's role begins with data preprocessing, where it is used to manipulate and prepare the raw data collected from sensors or cameras monitoring pet feed volume. Python libraries such as NumPy and OpenCV may be utilized for tasks such as image manipulation, resizing, and normalization.

TensorFlow

A powerful deep learning framework that is a pivotal component in building, training, and evaluating the Convolutional Neural Network (CNN) model. Leveraging TensorFlow's high-level APIs, such as Keras, simplifies the process of constructing complex neural network architectures, including CNNs. With TensorFlow, developers can define the layers, compile the model with appropriate optimizers and loss functions, and train the model on the preprocessed data efficiently. Additionally, TensorFlow's extensive documentation and community support provide insights into best practices and advanced functionalities, aiding in the optimization and fine-tuning of the CNN model for the specific task of estimating pet feed volume.

Furthermore, Python and TensorFlow facilitate the evaluation of the trained CNN model's performance using various metrics, such as accuracy and mean absolute error. This allows researchers to assess the model's effectiveness in accurately estimating the volume of pet food consumed based on visual inputs from the feeding station. Python's flexibility extends to model deployment, where frameworks like TensorFlow Serving or TensorFlow Lite enable the integration of the trained CNN model into production environments, allowing real-time inference on new data from pet feeding stations. Overall, Python and TensorFlow provide a robust ecosystem for implementing the "Pet Feed Volume Counter Using

Convolutional Neural Network" case study, from data preprocessing to model deployment, enabling efficient development and deployment of AI solutions for monitoring pet feeds volume.

Data Collection Instruments

For the "Pet Feed Volume Counter Using Convolutional Neural Network" case study, cameras are essential data collection instruments. High-resolution cameras, with specifications such as 1080p or 4K resolution, are strategically placed at pet feed stations to capture detailed images or video footage of superworms behavior. These cameras effectively capture the number of superworms present in a container. Considerations related to color and background play a crucial role in image accuracy. Superworms, being the primary focus, may share similar color depth with the background, potentially affecting the accuracy of images captured by cameras. To address this, you can employ techniques such as background subtraction to isolate superworms from the background. Additionally, color normalization can help adjust for variations caused by lighting conditions. By comprehensively analyzing visual data while considering the presence and characteristics of superworms in the storage environment, you can enhance the accuracy and reliability of your pet feed volume counting system implemented using Convolutional Neural Networks (CNNs).

Define Object Categories

In this case study the specific object that will be the main subject is superworms, also known as Zophobas morio larvae, are the primary objects of interest. They are characterized by their elongated bodies, segmented exoskeletons, and brownish-yellow color. This includes any part of the image that is not a superworm. The background can consist of various elements such as the container, substrate, or any other non-superworm objects. Occlusions and Overlapping Superworms can also be classified as a category in this subject, instances where superworms are partially or fully occluded by other superworms or objects within the image. This includes only a part of the superworm that may be visible and superworms may overlap, making it challenging to distinguish individual worms.

Figure 6. Zophobas morio larvae.

Selection of Data Collection Instruments

To effectively capture images of superworms high-resolution cameras will be used. These cameras will have a minimum resolution of 1080p to ensure that the images are clear and detailed. The captured images will be securely stored and easily accessible, initially saved on local devices such as SD cards or mobile phone storage. The core of the methodology is the use of Convolutional Neural Networks (CNN) for image recognition and object counting. The software components will include custom-trained models specifically designed for object detection and counting tasks. Additionally, libraries such as OpenCV will be utilized for image preprocessing tasks like resizing, normalization, and augmentation. Frameworks such as TensorFlow or Numpy will be employed to implement and train the CNN models.

Data Sources

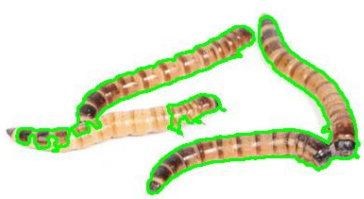
To gather data on superworms, several sources will be utilized. Existing insect datasets that may include superworms or similar species, though not exclusively focused on superworms, can provide relevant annotated images suitable for object detection. For more specific data, web scraping will be employed, using targeted image searches to find images of superworms with monochromatic or single-colored backgrounds for easier processing and analysis. Additionally, to supplement these datasets and scraped images, controlled image capture will be conducted. This involves taking photographs of superworms in controlled environments, ensuring a variety of scenarios while maintaining monochromatic or single-colored backgrounds to simplify the object detection process.

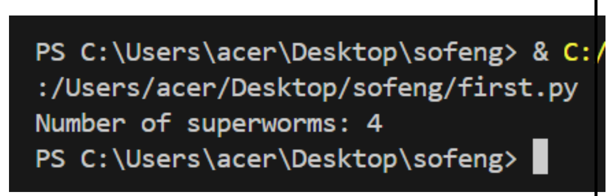
3. Verification and Testing

When it comes to the testing of the prototype, it is important to consider whether each part of the system is functioning as planned and whether the interactions between them are producing the right results. The testing phase will be centered on major areas such as image quality and preprocessing, the CNN model, the performance of the hardware, the real-time detection and counting, the user interface, and the overall system performance.

Firstly, the image quality and preprocessing were tested by capturing a number of high-resolution images under varying lighting and background environments. These images were processed to ensure they were in the right format, size, normalized, and augmented using libraries such as OpenCV and PIL. The quality and uniformity of these preprocessed images were assessed to ensure that they were fit for analysis. The CNN model was thoroughly trained to detect and count superworms in images with a high level of accuracy. This involved training the model on a labeled dataset and then testing the accuracy of the model on a different validation set. Evaluation measures, including accuracy, precision, recall, and F1-score, were used, and a cross-validation technique was employed to test the model's stability. The testing of the hardware performance entailed tracking the GPU, CPU, RAM, and SSD during the training and inference processes. This was to ensure that the system ran smoothly with minimal lag and did not overheat. Potential problems that could cause system constraints were highlighted and resolved to ensure the free flow of the system. The real-time monitoring capability of the system in counting the number of superworms was demonstrated by using a live camera that recorded the superworm activity. The CNN model was applied to this live feed, and the counts obtained from the video feed were compared with the manual counts to assess the accuracy. The response time of the system was also measured. Lastly, the functionality of the system was checked to determine whether it could handle all inputs, such as the overlapping of the superworms and the different orientations of the superworms, without any kind of failure. This entailed capturing images and videos of these difficult situations and then assessing the system's capacity to identify and quantify superworms correctly. To avoid such cases, error detection and recovery were incorporated to cater for any unexpected inputs.

Unit Test 1:

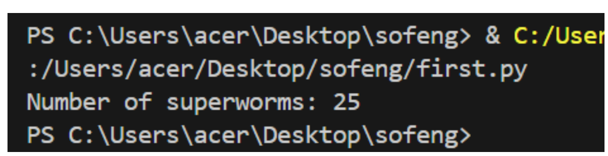




Unit Test ID	Component	Test Description	Expected Result	Status (Pass/Fail)	Date of Testing
1	Image Capture	Verify the camera that is used captures clear, detailed images of superworms under different lighting conditions.	Images are clear and focused.	Pass	Thu, 06 Jun 2024
2	Image Processing	Test images normalization and	Images are consistently	Pass	Thu, 06 Jun 2024

		augmentation using OpenCV and PIL libraries.	normalized and augmented correctly.		
3	CNN Model Training	Validate that the CNN model trains correctly on the annotated dataset.	Model trains without errors and shows improving accuracy over epochs.	Pass	Thu, 06 Jun 2024
4	Real-Time Detection	Test real-time detection and counting of superworms using a live camera feed.	System detects and counts superworms accurately with low latency.	Pass	Thu, 06 Jun 2024
5	Error Handling	Test system's robustness against unexpected inputs, such as overlapping superworms and different orientations.	System handles errors gracefully and maintains accurate counting.	Pass	Thu, 06 Jun 2024
6	Hardware Utilization	Monitor GPU, CPU, RAM, and SSD usage during training and inference.	System utilizes hardware efficiently without significant lag or overheating.	Pass	Thu, 06 Jun 2024
7	Data Storage	Ensure that the SSD stores images, datasets, and models efficiently.	Data is stored and retrieved quickly and without errors	Pass	Thu, 06 Jun 2024

Unit Test 2:

```
PS C:\Users\acer\Desktop\sofeng> & C:/User
:/Users/acer/Desktop/sofeng/first.py
Number of superworms: 25
PS C:\Users\acer\Desktop\sofeng>
```

Unit Test ID	Component	Test Description	Expected Result	Status (Pass/Fail)	Date of Testing
1	Image Capture	Verify the camera that is used captures clear, detailed images of superworms under different lighting conditions.	Images are clear and focused.	Pass	Thu, 06 Jun 2024
2	Image Processing	Test images normalization and augmentation using OpenCV and PIL libraries.	Images are consistently normalized and augmented correctly.	Pass	Thu, 06 Jun 2024
3	CNN Model Training	Validate that the CNN model trains correctly on the annotated dataset.	Model trains without errors and shows improving accuracy over epochs.	Pass	Thu, 06 Jun 2024
4	Real-Time Detection	Test real-time detection and counting of superworms using a live camera feed.	System detects and counts superworms accurately with low latency.	Pass	Thu, 06 Jun 2024
5	Error Handling	Test system's robustness against unexpected inputs, such as overlapping superworms and different orientations.	System handles errors gracefully and maintains accurate counting.	Pass	Thu, 06 Jun 2024
6	Hardware Utilization	Monitor GPU, CPU, RAM, and SSD usage during training and inference.	System utilizes hardware efficiently without significant lag or overheating.	Pass	Thu, 06 Jun 2024
7	Data Storage	Ensure that the SSD stores images, datasets, and models efficiently.	Data is stored and retrieved quickly and without errors	Pass	Thu, 06 Jun 2024

References

Heap. (n.d.). What is a Tech Stack? Heap. Retrieved from https://www.heap.io/topics/what-is-a-tech-stack

IBM. (n.d.). Machine Learning. IBM. Retrieved from https://www.ibm.com/topics/machine-learning

Circuit Digest. (n.d.). Arduino Weight Measurement Using Load Cell. Circuit Digest. Retrieved from https://circuitdigest.com/microcontroller-projects/arduino-weight-measurement-using-load-cell

Inflectra. (n.d.). Agile Scrum Methodology. Inflectra. Retrieved from https://www.inflectra.com/Solutions/Methodologies/Scrum.aspx

Li, G., Huang, Y., Chen, Z., Chesser Jr, G. D., Purswell, J. L., Linhoss, J., & Zhao, Y. (2021). Practices and applications of convolutional neural network-based computer vision systems in animal farming: A review. Sensors, 21(4), 1492. Retrieved from https://www.mdpi.com/1424-8220/21/4/1492

Joseph, R. V., Mohanty, A., Tyagi, S., Mishra, S., Satapathy, S. K., & Mohanty, S. N. (2022). A hybrid deep learning framework with CNN and Bi-directional LSTM for store item demand forecasting. Computers and Electrical Engineering, 103, 108358. Retrieved from: https://www.sciencedirect.com/science/article/abs/pii/S0045790622005754

Yanai, K., Tanno, R., & Okamoto, K. (2016, October). Efficient mobile implementation of a cnn-based object recognition system. In Proceedings of the 24th ACM international conference on Multimedia (pp. 362-366). Retrieved from: https://dl.acm.org/doi/abs/10.1145/2964284.2967243

[1] Sharma, N., Jain, V., & Mishra, A. (2018). An analysis of convolutional neural networks for image classification. Procedia computer science, 132, 377-384. Retrieved from: https://www.sciencedirect.com/science/article/pii/S1877050918309335

[2] Krizhevsky, A., & Hinton, G. (2009) "Learning multiple layers of features from tiny images.". Retrieved from: https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[3] Chauhan, R., Ghanshala, K. K., & Joshi, R. C. (2018, December). Convolutional neural network (CNN) for image detection and recognition. In 2018 first international conference on secure cyber computing and

communication (ICSCCC) (pp. 278-282). IEEE. Retrieved from: https://ieeexplore.ieee.org/abstract/document/8703316

[4] Zeiler, M. D. and Fergus, "Visualizing and understanding convolutional networks". European Conference on Computer Vision, vol 8689. Springer, Cham, pp. 818-833, 2014. Retrieved from: https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53

[5] Nguyen, H., Maclagan, S. J., Nguyen, T. D., Nguyen, T., Flemons, P., Andrews, K., ... & Phung, D. (2017, October). Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In 2017 IEEE international conference on data science and advanced Analytics (DSAA) (pp. 40-49). IEEE. Retrieved from: https://ieeexplore.ieee.org/abstract/document/8259762

Tan, M., Chao, W., Cheng, J. K., Zhou, M., Ma, Y., Jiang, X., ... & Feng, L. (2022). Animal detection and classification from camera trap images using different mainstream object detection architectures. Animals, 12(15), 1976. Retrieved from: https://www.mdpi.com/2076-2615/12/15/1976

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232. Retrieved from: https://ieeexplore.ieee.org/abstract/document/8627998

Ramalingam, B., Tun, T., Mohan, R. E., Gómez, B. F., Cheng, R., Balakrishnan, S., ... & Hayat, A. A. (2021). Ai enabled IoRt framework for rodent activity monitoring in a false ceiling environment. Sensors, 21(16), 5326. Retrieved from: https://www.mdpi.com/1424-8220/21/16/5326

Huberman-Spiegelglas, Inbar, and Raanan Fattal. "Single Image Object Counting and Localizing using Active-Learning." In Proceedings of the Winter Conference on Applications of Computer Vision (WACV), 2022. Available at: https://openaccess.thecvf.com/content/WACV2022/papers/Huberman-Spiegelglas_Single_Image_Object_Counting_and_Localizing_Using_Active-Learning_WACV_2022_paper.pdf

Santi Seguı, Oriol Pujol, and Jordi Vitria. "Learning to count with deep object features.". Retrieved from:

https://openaccess.thecvf.com/content_cvpr_workshops_2015/W03/papers/Segui_Learning_to_Count_2015_CVPR_paper.pdf

Uchida, S. (2013). "Image processing and recognition for biological images." Development, Growth & Differentiation, 55, 523-549. Available at: *https://onlinelibrary.wiley.com/doi/epdf/10.1111/dgd.12054*

Meiyin Wu and Li Chen, "Image recognition based on deep learning," 2015 Chinese Automation Congress (CAC), Wuhan, 2015, pp. 542-546. Available at: *https://doi.org/10.1109/CAC.2015.7382560*

Y. H. Liu, "Feature extraction and image recognition with Convolutional Neural Networks," Journal of Physics: Conference Series, vol. 1087, p. 062032, Sep. 2018. Available at: *https://iopscience.iop.org/article/10.1088/1742-6596/1087/6/062032/meta*

Binta Islam, S., Valles, D., Hibbitts, T. J., Ryberg, W. A., Walkup, D. K., & Forstner, M. R. (2023). Animal species recognition with deep convolutional neural networks from ecological camera trap images. Animals, 13(9), 1526. Retrieved from: *https://www.mdpi.com/2076-2615/13/9/1526*

Hijazi, S., Kumar, R., & Rowen, C. (2015). Using convolutional neural networks for image recognition. Cadence Design Systems Inc.: San Jose, CA, USA, 9(1). Retrieved from: *https://www.multimediadocs.com/assets/cadence_emea/documents/using_convolutional_neural_networks_for_image_recognition.pdf*