IMPLEMENTATION OF AI-POWERED SEARCH ALGORITHM TO GALLERY

TROY O. PRINCIPE

Thesis Manuscript Submitted to the Department of Computing and Library Information Science, College of Engineering and Information Technology, University of Southern Mindanao, Kabacan, Cotabato in Partial Fulfilment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

JUNE 2024

UNIVERSITY OF SOUTHERN MINDANAO
Kabacan, Cotabato
Philippines

APPROVAL OF THESIS MANUSCRIPT

Name	**TROY O. PRINCIPE**
Major	**N/A**
Degree Sought	**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**
Specialization	**N/A**
Thesis Title	**IMPLEMENTATION OF AI-POWERED SEARCH ALGORITHM TO GALLERY**

APPROVED BY THE GUIDANCE COMMITTEE

RALPH BUTCH S.GARIDAN
Adviser

Date

Co-Adviser
(Optional)

Date

Statistician
(Optional)

Date

NOR-AINE M. CORPUZ
Department Research Coordinator

Date

ELIZABETH R. GENOTIVA
Department Chairperson

Date

NOR-AINE M. CORPUZ
College Research Coordinator

Date

MARICEL G. DAYADAY, DTE
Dean

Date

Study No DCLIS-CS-2022-
Index No: MC-CEIT-008168
Recorded by: nmc

RECORDED:

LYDIA C. PASCUAL
Director for Research and Development Office

Date

Recorded by: __________

USM-EDR-F05-Rev.4.2020.11.16

	UNIVERSITY OF SOUTHERN MINDANAO Kabacan, Cotabato Philippines	
	ACCEPTANCE OF THESIS	

The thesis attached hereto, entitled "**IMPLEMENTATION OF AI-POWERED SEARCH ALGORITHM TO GALLERY**" prepared and submitted by **TROY O. PRINCIPE** in partial fulfilment of the requirements for the degree of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE** is hereby accepted.

RALPH BUTCH S.GARIDAN
Adviser

Date

NOR-AINE M. CORPUZ
College Research Coordinator

Date

USM-EDR-F10-Rev.4.2022.10.18

BIOGRAPHICAL DATA

The author of this undergraduate thesis is Troy O. Principe, born on January 27, 2000, in Libungan, Cotabato, Philippines. Having graduated from high school in 2018, Troy pursued his passion for technology by enrolling in the University of Southern Mindanao, where he is currently pursuing a Bachelor of Science in Computer Science. His interest in computers blossomed during his junior high school years, where he graduated with special awards from Padura-Espabo High School. Subsequently, he specialized in ICT during his senior high school studies at Libungan National High School, culminating in the acquisition of his TESDA National Certificate II. This achievement further fueled his aspiration to pursue a degree in computer and information technology.

During his academic journey, Troy completed his on-the-job training at the University of Southern Mindanao's Graduate School, where he gained practical experience in the field. Driven by his passion and dedication, Troy aims to forge a career in the dynamic and ever-evolving realm of computer and information technology, leveraging his academic background and hands-on experience to contribute meaningfully to the field.

TROY O. PRINCIPE
Researcher

ACKNOWLEDGEMENT

I am profoundly grateful for the journey of self-discovery and growth that this thesis represents. It is with immense appreciation that I acknowledge the dedication, perseverance, and hard work I have invested in this endeavor. Through moments of challenge and setbacks, I remained steadfast in my commitment to see this study to fruition, driven by a relentless pursuit of knowledge and academic excellence.

My heartfelt gratitude extends to my family for their unwavering love, encouragement, and unwavering support throughout my undergraduate education, particularly my sister, May P. Lagarnia, who has provided everything for me in this college journey. Their sacrifices and continuous inspiration have been instrumental in my academic journey, providing me with the courage and motivation to overcome obstacles and reach new heights. I am also grateful to my friends and classmates for their support and cooperation during the course of this project. Their wise counsel and constructive criticism have played a pivotal role in shaping my work and methodology, enriching the overall research experience.

I am indebted to Mr. Ralph Butch S. Garidan, my esteemed thesis adviser, for his invaluable guidance, patience, and encouragement throughout the research process. His insights and constructive feedback have significantly

contributed to the refinement of my ideas and the enhancement of the thesis's quality.

Lastly, I offer my deepest gratitude to the Lord God for His grace and blessings, which have sustained me through the challenges and trials encountered along this academic journey. It is my fervent hope that this work will be pleasing in His sight and make a modest contribution to the advancement of knowledge in the field.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ABSTRACT

PRINCIPE, TROY OSTIQUE, 2024. Implementation of AI-powered Search Algorithm to Gallery. Bachelor of Science in Computer Science Thesis. College of Engineering and Information Technology, University of Southern Mindanao, Kabacan, Cotabato. 58 pp

Adviser: **RALPH BUTCH S. GARIDAN**

In contemporary Android gallery applications, efficiently searching and retrieving images and videos from extensive collections poses a significant challenge. Traditional keyword-based search algorithms exhibit limitations in contextualizing search terms and often yield irrelevant results. This research addresses these challenges by integrating artificial intelligence (AI) into search algorithms, specifically using object detection algorithms like YOLO (You Only Look Once). The primary aim is to enhance user experience by implementing an AI-powered search algorithm that leverages machine learning to recognize objects in multimedia content, providing more accurate and relevant search results based on content rather than filenames. The study involved developing a mobile gallery application for Android devices, integrating the YOLO object detection algorithm, and evaluating the AI-powered search algorithm's accuracy and effectiveness. Significant findings from user testing with 30 participants revealed that ease of use was well-received, with 23 users rating it as either "Very Satisfied" or "Satisfied." The user interface received positive feedback from 20 users, while the accuracy of search results was praised by

23 users. However, search speed was a notable drawback, with 25 users expressing dissatisfaction. Confusion matrix analysis highlighted the strengths and weaknesses of the YOLOv5 model, showing high true positive rates in several object classes but identifying areas needing improvement. To further enhance functionality, several key recommendations are proposed: implementing background inference for object detection, exploring newer YOLO versions, storing scan results in a JSON file, integrating natural language processing capabilities, incorporating voice search capabilities, and exploring on-device optimization techniques. These recommendations aim to reduce computational load, improve accuracy, enhance user interaction, and significantly boost overall application performance and user experience.

Keywords: algorithm, Android, AI, confusion matrix, gallery, machine learning, NLP, YOLO

INTRODUCTION

In the digital age, the task of searching and retrieving images and videos from a vast collection can often become burdensome, especially in Android gallery applications where numerous files are stored. Traditional keyword-based search algorithms have served as a solution to this issue, yet they possess inherent limitations. These include their inability to contextualize search terms or discern multiple meanings, resulting in either irrelevant outcomes (false positives) or the failure to retrieve related materials (false negatives) (RightsDirect, 2021). Such limitations hamper users' ability to efficiently locate desired content, thereby impacting the overall user experience of applications reliant on search functionalities.

To address these challenges within Android gallery applications, this study explores the integration of artificial intelligence (AI), specifically object detection algorithms like YOLO, into search algorithms. Such integration has the potential to vastly enhance user experiences by furnishing more accurate and pertinent search results. This is achieved through machine learning techniques capable of recognizing objects in images and videos, enabling algorithms to present results based on relevance rather than mere filename matching. Notably, modern object detection algorithms like YOLO (You Only Look Once) have demonstrated swift and precise identification of objects in multimedia content.

The overarching aim of this study was to refine the user experience of mobile gallery applications by furnishing a swifter and more precise method for searching and retrieving images. Additionally, it seeks to contribute to the fields of artificial intelligence and mobile application development by exploring the potential of AI-powered search algorithms in mobile gallery applications. Specifically, the study aimed to develop a mobile gallery application facilitating storage and access to images and videos on mobile devices, integrate an AI-powered search algorithm into the mobile gallery application, utilize the YOLO object detection algorithm to train the AI-powered search algorithm, evaluate the overall accuracy of the AI-powered search algorithm in the mobile gallery application through user testing, and employ a confusion matrix to analyze the accuracy of the AI-powered search algorithm.

This study holds significance in its potential to augment the search functionality of Android gallery applications through the implementation of an AI-powered search algorithm. Presently, search algorithms in Android gallery applications solely yield results based on filenames, posing limitations that hinder users from efficiently locating desired content, especially within large albums. In contrast, the integration of an AI-powered search algorithm, leveraging machine learning to identify the relevance of multimedia content to user queries, has the potential to significantly enhance the search experience.

Moreover, beyond enhancing Android gallery applications, this study contributes to the broader field of Artificial Intelligence research. By offering valuable data and insights, it informs future researchers on the applications of AI algorithms in diverse contexts. Furthermore, serving as a proof-of-concept for AI-powered search algorithms in mobile applications, it may inspire future studies on enhancing user experiences in other mobile applications.

Definition of Terms

For the benefit of the readers, terms are operationally defined below:

AI - Artificial Intelligence employs the use of technology such as computers and machines to imitate the cognitive functions of the human brain, including the ability to solve problems and make decisions.

AI-powered - AI-powered is a systems or applications that use artificial intelligence to perform tasks or make decisions. These systems can be trained to perform a wide range of tasks and are often used to improve efficiency and accuracy by automating tasks that would otherwise be done by humans.

Algorithm - An algorithm is a systematic procedure or method for achieving a specific outcome or goal, by executing a series of well-defined and logical steps in a precise sequence. It is essentially a step-by-step plan for solving a problem or completing a task.

Android - Android, developed by Google, is a Linux-based operating system that primarily powers smartphones and tablets. It offers a vast array of features, including customization options, and a large app market.

Application - An application, or app, is a software program designed to perform a specific function or set of functions. Applications are typically designed to run on a specific platform, such as a computer operating system, a mobile device, or the web.

Confusion Matrix - A confusion matrix is a tool used to evaluate the performance of a classification algorithm in a multi-class problem. It is a tabular representation that compares the predicted class labels against the true class labels, and provides insights into the accuracy and error of the algorithm.

Gallery - The mobile gallery application, often pre-installed on Android devices, is a user-friendly software that allows the management, organization and viewing of images and videos on a smartphone.

NLP - It enables computers to instantaneously complete tasks that might otherwise require a person to perform them, like language translation, text summarization, question answering, and more.

Search Algorithm - It is a specific type of algorithm that is designed to find a particular item or group of items in a dataset. Search algorithms are used in many different contexts, including databases, the internet, and other information systems.

YOLO - short for "You Only Look Once," is an efficient algorithm for identifying and locating various objects within an image. This method employs real-time object detection by treating it as a regression problem, and subsequently generating class probability predictions for the identified objects.

Conceptual Framework

The conceptual framework delineates the study's structure, focusing on integrating AI into search algorithms within Android gallery applications to enhance user experiences in accessing multimedia content, encompassing the development of a mobile gallery application, algorithm training, user testing, and evaluation, and accuracy analysis using a confusion matrix as shown in figure 1.

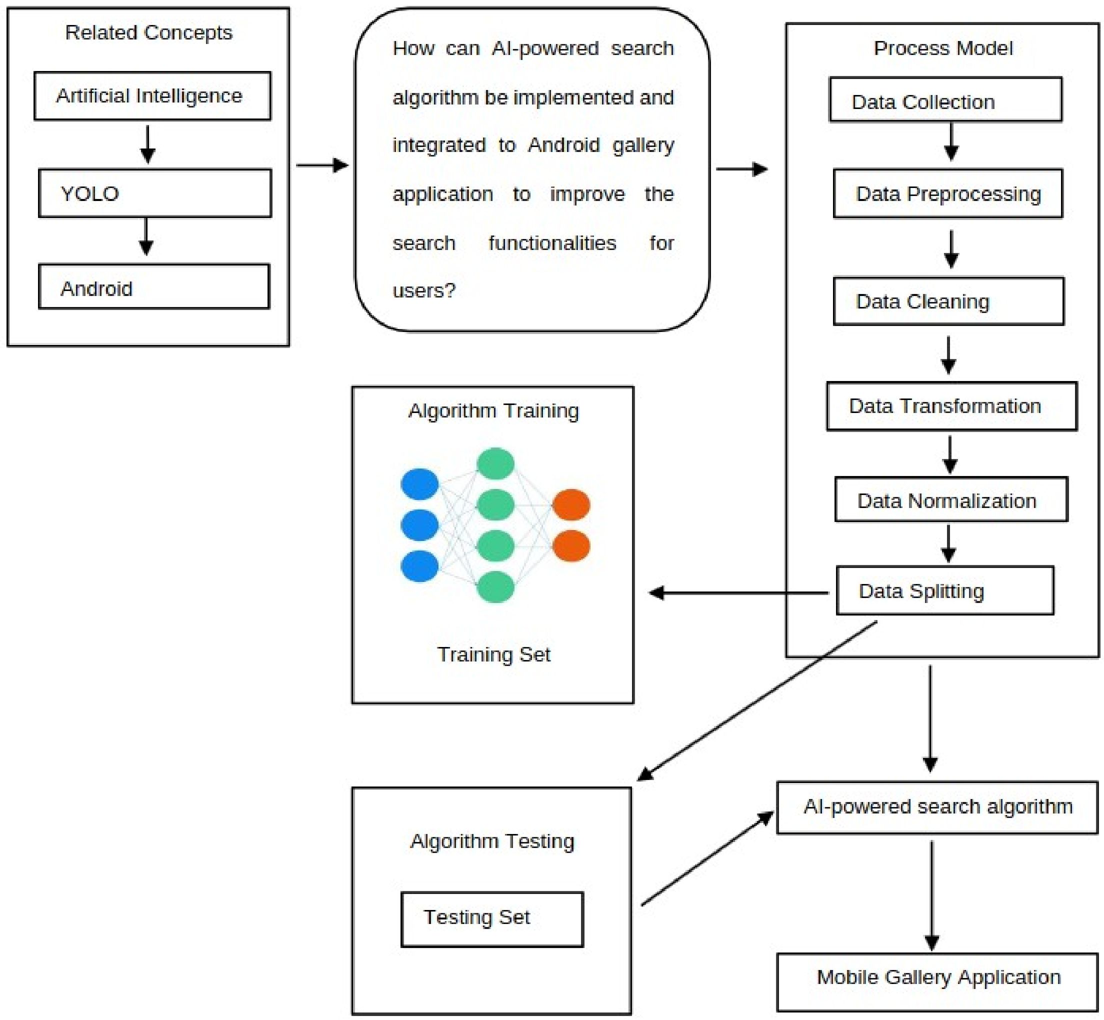

Figure 1. Conceptual Framework of the study.

REVIEW OF RELATED LITERATURE

This chapter presents a review of literature on search algorithms, artificial intelligence, and AI-powered systems which are relevant to the current research study. The review aims to provide context and background information on these topics and to highlight the ways in which previous researches have informed and influenced the current investigation; to provide a comprehensive overview of the state of the field and to identify key themes and trends that are relevant; and to contribute to the broader understanding of search algorithms, artificial intelligence, and AI-powered systems and to identify areas where further research is needed.

AI-powered Systems

There is a significant amount of literature available on AI-powered systems, with researchers examining a wide range of topics related to their development, deployment, and impact. One notable publication in this field is a research conducted by Yu et al. (2018), examined the potential for artificial intelligence (AI) to enhance learning and research in Massive Open Online Courses (MOOCs). The authors proposed several ways in which AI could be used to improve MOOCs, including the use of knowledge representation tools to allow students to tailor their learning experience to their own needs, the use of optimization techniques to match community teaching assistants with

MOOC mediation tasks, and the implementation of virtual learning companions with human-like traits such as curiosity and emotions.

In the article "Building the AI-powered organization" by Fountaine et al. (2019), the authors discuss the common mistake of viewing artificial intelligence (AI) as a quick fix that can bring immediate returns. They argue that this approach is not effective and suggest instead focusing on AI as a strategic capability that requires a long-term commitment to building the necessary data infrastructure, expertise, and model development. The authors also recommend using AI to solve more general business problems rather than applying it to specific, isolated issues.

The research conducted by Thakkar et al. (2020) explored the perceptions and practices surrounding automation in the future of work among vocational technicians in Bangalore, India. The authors found that these technicians, who are highly vulnerable to the potential impact of automation on their jobs, were largely unaware of the growth of automation and expressed a vision for the future of work that was in line with their values. They also reported feeling excluded by current technological platforms for skill development and job seeking. The authors suggest that there are opportunities for technology industry and policy makers to build a future of work that is inclusive of vulnerable communities.

A review of the literature on search algorithms, artificial intelligence, and AI-powered systems has shown that these technologies have had a significant impact on various fields and industries. In particular, search algorithms have been widely used to improve the efficiency and accuracy of information retrieval, and have been applied in a variety of contexts such as web search, database search, and document search. Artificial intelligence, on the other hand, has been used to develop systems that can simulate human-like intelligence and decision-making capabilities. AI-powered systems have been applied in various domains including natural language processing, image recognition, and autonomous systems.

The literatures on these topics has inspired the researcher to conduct this study by providing a strong foundation of knowledge and understanding on the capabilities and potential applications of search algorithms, artificial intelligence, and AI-powered systems. This knowledge has helped shape the direction and focus of this study by providing a clear understanding of the current state of the field and the potential areas for further research and development.

Artificial Intelligence

Below are some of the literatures that discussed the use of deep learning-based approaches, which are a type of AI that involves training large neural networks on large datasets to recognize patterns and make decisions. These approaches have been shown to be highly effective and efficient for object detection, object recognition, and image classification, and have achieved state-of-the-art performance on a wide range of tasks and datasets.

One of the most notable literature is the work by Redmon et al. (2016), the researcher conducted research on a new method for object detection called YOLO (You Only Look Once). YOLO approaches object detection as a regression problem, predicting bounding boxes and class probabilities directly from full images. This method allows for the optimization of the entire detection process end-to-end, resulting in a highly efficient system. The base YOLO model was able to process images at a rate of 45 frames per second, while the smaller version, Fast YOLO, was able to process 155 frames per second with a high level of accuracy. YOLO was also found to be less likely to predict false positives on background compared to other state-of-the-art detection systems, although it did have more localization errors.

Zhao et al. (2019), provides a comprehensive overview of deep learning-based object detection frameworks, including typical generic object detection architectures and specific tasks such as salient object detection, face detection, and pedestrian detection. The review also compares various methods through experimental analyses and suggests promising directions for future work in this field. Overall, the review highlights the significant advancements made in object detection through the use of deep learning, and the potential for continued progress in this area.

These and other studies that concentrated on image recognition will be useful in guiding this research. He et al. (2016) conducted a study that dealt with image recognition. They offer a residual learning architecture to make it simpler to train networks that are far deeper than those previously employed. Similar research was conducted by Wang et al. (2017) on convolutional neural networks that use attention mechanisms and cutting-edge feed forward network architecture in an end-to-end training scheme. Figurnov et al. (2017) suggests a Residual Network-based deep learning architecture that dynamically modifies the number of executed layers for the various regions of the image.

Duta et al. (2021) found that a robust form of convolutional neural network (CNN) architecture known as residual networks (ResNets) is well-liked and applied to a variety of tasks. Convolutional networks can be significantly deeper, more precise, and easier to train if they have shorter connections between layers that are close to the input and those that are close to the output (Huang et al., 2017).

Search Algorithms

In the context of search algorithms, there have been a number of recent studies that have explored the various aspects of this field. The following literatures are some of the most relevant to this topic. In the study by Wang et. al (2019), it was discovered that incorporating context information into both parts of the proposed model and using neural networks to accurately classify the cost functions of the A* algorithm towards the personalized route recommendation (PRR) task can produce more precise and customized route recommendations. The efficacy and robustness of the proposed model were demonstrated through experiments on three real-world datasets.

Yu et al. (2018) proposed a multiple learning backtracking search algorithm (MLBSA) that combines the exploration abilities of individuals that learn from current and historical population information with the convergence speed of individuals that acquire knowledge from those of the feature subset in the existing population. The MLBSA was tested on the parameters identification problems of three photovoltaic models: single diode, double diode, and photovoltaic module. The results of the experiments demonstrated that the MLBSA was more accurate, reliable, and computationally efficient than other state-of-the-art algorithms.

A search algorithm's suitability is frequently determined by the data structure being searched. It could also take into account previous knowledge of the data. Whoever it may be—humans or AI—must consider all viable paths —ncluding all possible outcomes—to reach the goal state, if one exists from the initial state. Similar to this, if a specific goal state exists, AI systems use a variety of search algorithms to find it (Sriniketh, 2021).

Appriliant (2021) made a comparison between Breadth-First Search and Depth-First Search for tree transversal. He found out that the use of DFS is more advised when the data structure is more tightly packed and has a deep structure. On the other hand, BFS will be more effective if the data structure has a tendency to be dispersed and isn't deep enough. The DFS algorithm traverses through a tree or graph by exploring one branch as deeply as possible before retracing its steps to examine alternative routes. The method begins by traversing down one path and continues until it reaches the end of the branch, then it backtracks to the first point of deviation and proceeds to explore other possible paths (Niketik, 2021). The benefit of this approach is that it only needs a small amount of memory because only the nodes on the active path are kept in memory (Ginting & Sembiring, 2019). Unlike DFS, BFS doesn't aggressively go though one branch until it reaches the end, rather it visits all the unvisited neighbors of that node before proceeding to another node (Popovic, 2020).

Understanding how the algorithm works and learning to use it to your advantage can help you increase website traffic, build better relationships with potential customers, and ultimately grow your business. According to Ofiwe (2021), the mystery surrounding Google's search algorithm can be both frustrating and fascinating for a digital marketer, content marketer, or SEO. Without a doubt, one of the most important technologies ever developed is Google's search algorithm. Google, with its staggering 5.6 billion daily searches, exerts a significant influence on the world and on businesses alike. This search engine giant has a far-reaching impact on various industries and has become a fundamental part of daily life for many people (Widmer, 2022). On the other hand, Bing has a transparent approach to website ranking by detailing the factors it takes into account. These include relevance, quality, credibility, user engagement, freshness, location, and page load speed (Schwartz, 2020).

Currently, the major objective of search engine algorithms is to deliver the most pertinent results for each individual search query. According to a study 12 conducted by Halavais (2017) there is a growing concern about how society is being impacted by algorithms, which operate beneath the surface of our online interactions. That being said, it also means that these algorithms are unknowingly influencing our knowledge and views about the world in which we live.

METHODOLOGY

This section details the methodology used to develop and evaluate the AI-powered search algorithm for the Android gallery application. Key stages include selecting an open-source gallery app, integrating and optimizing the YOLOv5 object detection model, implementing the AI-powered search functionality, and conducting thorough user testing. Each phase aims to ensure the algorithm provides accurate and contextually relevant search results, enhancing overall user experience. The following subsections describe the specific techniques and tools employed in each phase.

Research Design

This research involved a combination of experimental and observational methods, as the researcher needed to design and build the application, as well as implement and test the AI-powered search algorithm. One key aspect of the research design was the development of the Android gallery application and the implementation of the AI-powered search algorithm.

Data Collection

In the process of data collection, the researcher leveraged publicly available open-source datasets to effectively train the YOLOv5 model, a state-of-the-art machine learning algorithm. The datasets were meticulously organized into three distinct segments: training, validation, and testing. Each segment played a crucial role in the model's development and refinement. The training dataset was primarily used to teach the model how to accurately identify and classify different objects. The validation dataset served as a tool for tuning the model's parameters and for preventing overfitting by providing a reliable way to validate the model’s performance during the training phase. Finally, the testing dataset was employed to assess the model's accuracy and effectiveness in a simulation of real-world conditions, thus providing an unbiased evaluation of its capabilities.

To facilitate the annotation of these datasets, the researcher utilized Roboflow, a robust platform known for its efficiency in handling and annotating image data. Roboflow enabled the precise marking of objects within images, which is critical for training accurate object detection models like YOLOv5. This annotation process not only helped in identifying the objects within the dataset images but also in improving the overall precision of the model by providing clear, consistent labels across the entire dataset.

Participants and Materials

In this research, an open-source Android gallery was utilized as the foundational framework, onto which the AI-powered search algorithm was implemented. The study is meticulously developed and tested by the researcher within a controlled environment. As such, it does not require additional participants to complete the research protocol. Nevertheless, the findings of this investigation may hold significant relevance for practical applications.

The computational infrastructure employed in this research comprises a laptop equipped with Arch Linux, by the way, featuring an Intel Core i5 12th generation processor and 32 gigabytes of RAM. For testing purposes, an Android 12 mobile phone is utilized. Notably, the researcher leverages Google Colab, a cloud-based platform, for training the YOLOv5 machine learning algorithm.

The development of the Android Gallery application is undertaken in Android Studio, with Kotlin serving as the primary programming language. This meticulous approach to both hardware and software selection underscores the rigor and precision inherent in the study's methodology.

Validity

The mobile gallery application underwent validation and testing overseen by the researcher's thesis adviser, ensuring alignment with established standards of research validity. Collaborative efforts between the researcher and adviser enabled meticulous examination of the application's functionality, usability, and performance metrics. Diverse testing scenarios and user feedback were incorporated to enhance comprehensiveness and reliability, fostering trustworthiness and generalizability of the study's findings in mobile application development and usability assessment.

Development Methodology

a. **Identification of the Problem**

Traditional keyword-based search algorithms in Android gallery applications had limitations that could negatively impact the user experience by returning irrelevant results or failing to turn up related materials. These limitations made it difficult for users to efficiently locate and access desired content in large datasets with a diverse range of content, leading to frustration and decreased productivity.

To address this problem, the researcher aimed to investigate the potential of integrating artificial intelligence, specifically object detection algorithms like YOLOv5, into search algorithms for Android gallery applications. By utilizing machine learning techniques to recognize and identify relevant content in images and videos, AI-powered search algorithms had the potential to provide more accurate and relevant search results, thereby improving the user experience and productivity.

b. Analysis of the Problem

i. Input Requirement

The input requirements for this study included keywords that were to be input in the search box to initiate the search; datasets of images and videos that were used to train the AI-powered search algorithm; the necessary software and hardware resources, such as the Kotlin programming language and its libraries for developing the Android gallery application and the AI-powered search algorithm; and laptops and Android phones for testing and evaluation.

ii. **Output Requirement**

The primary output was the display of accurate and relevant search results based on the keywords entered by the user. The system needed to efficiently retrieve and display images and videos that closely matched the search terms.

iii. Implementation

The implementation of the YOLOv5 model into an open-source Android gallery presented both challenges and opportunities. Integrating the model into the existing framework required a meticulous approach due to limited resources and guidance available for loading YOLOv5 using PyTorch libraries. Despite these hurdles, the decision to replace the conventional fuzzywuzzy search algorithm with YOLOv5 proved transformative. Leveraging YOLOv5's object detection capabilities, the Android gallery now efficiently scans the filesystem to retrieve images relevant to the search query, significantly enhancing search accuracy and user experience. This implementation not only underscores the versatility of YOLOv5 in real-world applications but also highlights the ingenuity required to overcome technical obstacles and harness the full potential of advanced AI models in mobile platforms.

Training Phase

This section outlines the steps involved in training the YOLOv5 model. The training dataset comprised 669 images, with each of the 80 classes represented by 5 images. The validation dataset consisted of 160 images, providing a smaller, separate collection of data used to evaluate the effectiveness and accuracy of the model during the training process, without adjusting the model itself. Finally, the testing dataset included 80 images and was used to assess the model's performance after training was complete. This separate dataset helped ensure that the model's predictive capabilities were reliable and effective in real-world scenarios, not just under controlled training conditions.

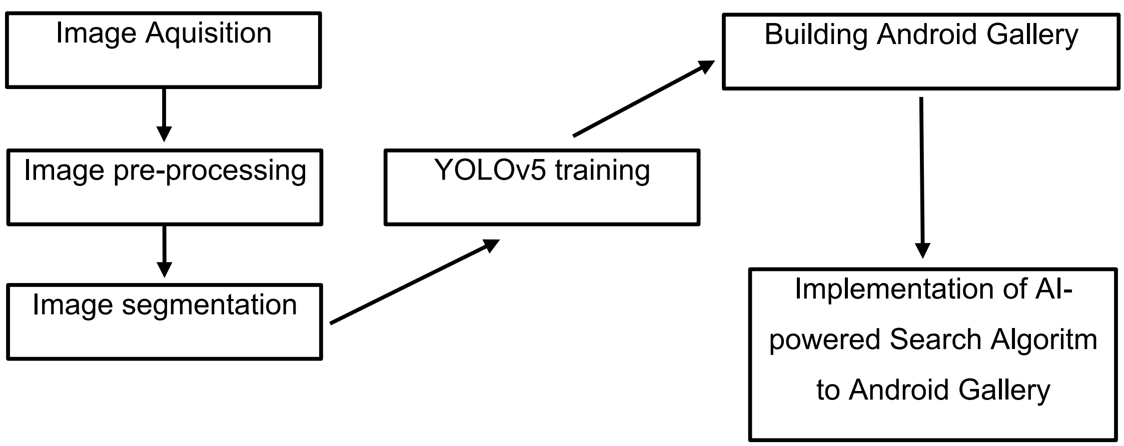

Figure 2. Workflow of the study

Image Acquisition

The image acquisition phase was a critical component of the research, serving as the foundation for the subsequent training of the YOLOv5 model. The process involved several meticulously planned steps to ensure a robust and diverse dataset that could effectively train the AI to recognize and classify a wide array of objects. The first step involved identifying diverse sources from which to collect images. These sources included publicly accessible image databases, online repositories that could provide access to image collections. The aim was to gather a wide variety of images representing different environments, lighting conditions, and perspectives to enhance the model's robustness.

Pre-processing

The image preprocessing phase is pivotal in preparing the acquired images for effective training of the YOLOv5 model. This phase involves several steps designed to standardize the input data, enhance image quality, and augment the dataset to improve the model’s robustness and accuracy. All images were resized to a uniform dimension, ensuring consistency across the dataset. This step is crucial because YOLOv5 requires input images of the same size to efficiently process and analyze the data. To increase the robustness of the model against overfitting and to improve its ability to generalize across different environments, data augmentation techniques were employed. These included random rotations, horizontal flipping, brightness adjustments, and scaling. These transformations mimic various real-world conditions, thereby providing a more comprehensive training experience. Figure 3 shows the process of image annotation, illustrating the steps involved in labeling images with relevant object classes.

Figure 3. Image Annotation

Segmentation

The largest portion of the dataset was allocated to training. This subset includes a wide variety of images encompassing all classes that the YOLOv5 model meant to detect. The training dataset is used to adjust the weights of the neural network, effectively "teaching" the model how to identify and classify different objects. The validation dataset is used to tune the hyperparameters of the model and to prevent overfitting. It acts as a checkpoint to evaluate the model during the training phase without using the test data. This subset is crucial for verifying that the model generalizes well to new, unseen data. Finally, the testing dataset is used only after the model has been trained and validated. This subset is crucial for assessing the final model's performance and its ability to generalize to new data. The testing dataset is completely independent of the training and validation datasets. Figure 4 shows the process of dataset segmentation, detailing how the complete dataset is divided into distinct subsets for training, validation, and testing purposes.

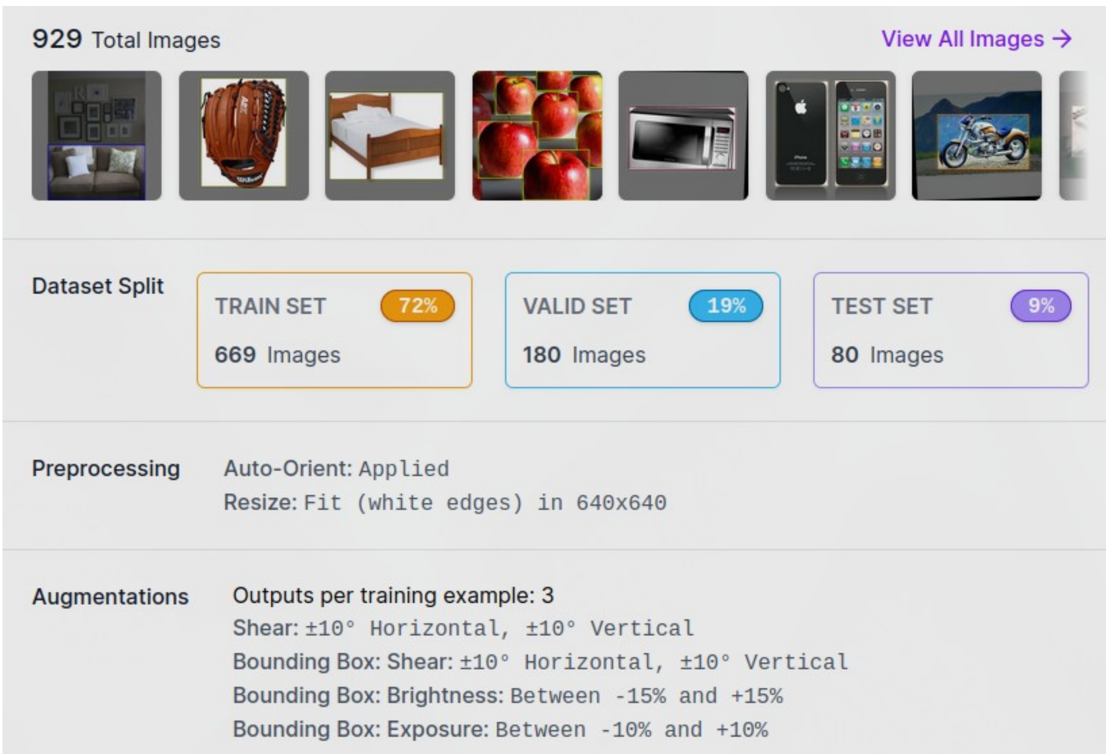

Figure 4. Segmented Datasets

Feature Extraction

In the feature extraction phase of the YOLOv5 model training, the raw image data undergoes a series of convolutional operations within the neural network architecture to extract meaningful features. This process involves passing the images through multiple layers of convolutional filters, pooling operations, and activation functions, which progressively capture hierarchical representations of the input images. These extracted features encode various visual patterns, textures, and shapes present in the images, enabling the model to discern relevant objects and their spatial relationships within the scene. Through this iterative feature extraction process, the YOLOv5 model learns to transform the raw pixel values into higher-level representations that facilitate accurate object detection and classification during inference.

Trained Model

The trained YOLOv5 model encapsulates the culmination of the training process, embodying a sophisticated neural network architecture that has been fine-tuned to recognize and classify objects within images with high accuracy and efficiency. Through extensive exposure to labeled image data during the training phase, the model has learned to extract intricate features and patterns, enabling it to make informed predictions about the presence and locations of objects of interest. With optimized parameters and learned weights, the trained model is capable of real-time object detection, offering a powerful tool for various applications, from surveillance and autonomous vehicles to image classification and content recommendation systems.

Loading the YOLOv5 Model

The seamless integration of the YOLOv5 model into the Android gallery application involves a series of steps, ensuring its efficient functionality. Within the SearchViewModel class, serving as the core for search queries and media retrieval, loading the YOLOv5 model is fundamental. This process commences with the initialization of a lazy property, yoloModel, leveraging the loadYoloModel() function. This function orchestrates the retrieval of the model file, "yolov5s.torchscript.ptl," from the assets directory, facilitated by PyTorch's LiteModuleLoader. Furthermore, the loadClasses() function is employed to load the classes.txt file, containing labels for detected objects. These classes are then assigned to the PrePostProcessor.mClasses property, facilitating subsequent object detection processes. This meticulous loading mechanism ensures the YOLOv5 model is seamlessly accessible for detecting objects within images and videos showcased within the gallery application, owing to the robust functionalities provided by PyTorch libraries.

Post-Processing YOLOv5 Inference Results

The PrePostProcessor class within the Android gallery application's Search feature serves as the post-processing stage for YOLOv5 model inference results. This class houses methods designed to refine the raw output, particularly focusing on non-maximum suppression (NMS) predictions. The nonMaxSuppression method is pivotal, as it filters out redundant bounding boxes by iteratively selecting those with the highest confidence scores while removing overlapping boxes beyond a specified threshold. Concurrently, the IOU method calculates the intersection-over-union (IOU) overlap between bounding boxes, aiding in the refinement process. Finally, the outputsToNMSPredictions method consolidates these procedures by converting the raw model outputs into refined predictions, adjusting bounding box coordinates to align with the input image size and applying NMS to ensure the accuracy and reliability of object detection within the Android gallery application.

Querying and Filtering Media

Upon initialization, it loads a YOLOv5 model, an advanced deep learning architecture recognized for its precision in object detection tasks. When a user initiates a search, the `queryMedia()` function triggers the retrieval of media items from the application's data source. Following this initial step, each media item undergoes meticulous scrutiny through the `parseQuery()` function. Here, the ViewModel first processes the user's query, identifying key elements or labels. Subsequently, leveraging the YOLOv5 model, the function meticulously examines each media file, scanning for objects aligning with the identified query elements. Crucially, a predefined confidence threshold is applied to ensure the relevance and accuracy of the detected objects.

With the query elements identified and object detection underway, the ViewModel proceeds to refine the results. The YOLOv5 model executes inference on each media item, discerning relevant objects within images. This inference process is vital for pinpointing objects that closely match the user's search query. Once objects are detected, the ViewModel applies additional filters, ensuring that only objects surpassing the confidence threshold are considered relevant. Through this meticulous process of query processing, inference execution, and result refinement, irrelevant media files are effectively filtered out. Finally, the ViewModel updates its internal state, reflecting the curated collection of media items, and broadcasts this information via a state flow.

User Testing

The Table 1 shows the questionaire that will be used for user testing, it will be used to evaluate the following; Ease of Use, User Interface, Accuracy of Search Results, Search Speed, and Overall Experience.

Question	Very satisfied	Satisfied	Neutral	Dissatisfied	Very dissatisfied
How would you rate the ease of use of the AI-powered search feature in the gallery application?					
How would you rate the user interface and design of the search feature within the gallery application?					
How satisfied are you with the speed at which the AI-powered search retrieves the images?					

How accurately did the AI-powered search feature retrieve images containing the object you searched for?					
Overall, how satisfied are you with the AI-powered search feature in the gallery application?					

Software Requirements

The Table 1 shows the minimum and recommended software requirements for developing the Android gallery. These are required for the framework to run without any problems.

Table 1. Software Requirements.

Software	Specification	
	Minimum	**Recommended**
Android Studio	2023.1.1	2023.3.1
Operating System	Arch Linux	Arch Linux
System Type	64-bit	64-bit
Android	11	14

Hardware Requirements

The Table 2 shows the minimum and recommended hardware requirements for developing the Android gallery.

Table 2. Hardware Requirements.

Hardware	**Specification**	
	Minimum	**Recommended**
Processor	Intel® Core™ i5 10th gen	Intel® Core™ i7 12th gen
Ram	16 GB	32 GB
SSD	256-512 GB	512 GB-1TB

RESULTS AND DISCUSSION

This section presents the comprehensive outcomes derived from the meticulous implementation of the AI-powered search algorithm within the Android gallery application, meticulously aligning with the outlined research objectives. The discussion not only encapsulates the intricate development phases of the mobile gallery application and the seamless integration of the AI-powered search algorithm but also delves into the rigorous performance evaluation, augmented by a meticulous analysis employing the confusion matrix.

Develop a Mobile Gallery Application

In pursuit of our goal, we utilized an open-source Android gallery application as the foundational framework for implementing the AI-powered search algorithm. By leveraging an existing gallery application, we can expedite the development process and focus our efforts on integrating the AI-powered search functionality seamlessly. This approach allows us to capitalize on the existing features of the gallery application, such as image and video storage and retrieval functionalities, while also providing a familiar user interface for our target audience. Additionally, building upon an open-source platform enables collaboration with the developer community, facilitating enhancements and updates to the application over time. Overall, utilizing an open-source Android gallery application as the base foundation provides a solid framework for implementing the AI-powered search algorithm efficiently and effectively.

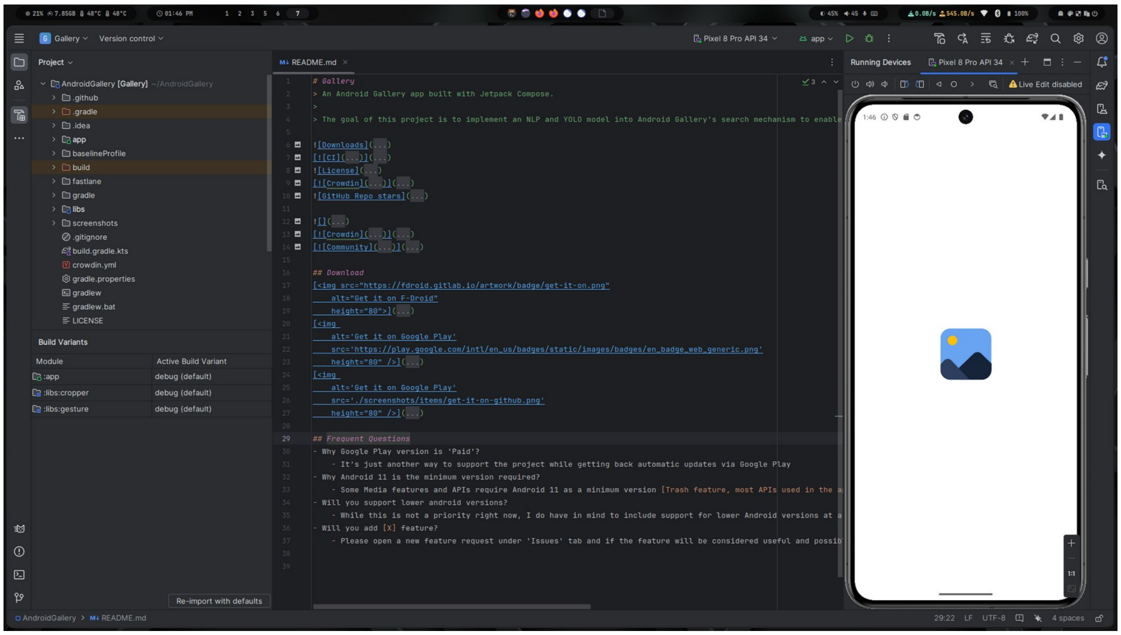

Figure 5. Mobile Gallery Application Development

Training the YOLOv5 Model

The AI-powered search algorithm's development entails training the YOLOv5 model for precise object detection, crucial for enhancing the gallery application's search functionality. This crucial task is accomplished through Ultralytics's YOLOv5 Google Colab notebook, meticulously designed to streamline the training process by providing essential dependencies required for training the YOLOv5 model. Leveraging the robust infrastructure and computational resources offered by Google Colab, which hosts a cloud-based Jupyter notebook environment, ensures a seamless and scalable approach to training the YOLOv5 model.

Throughout the training process, key parameters are set to optimize model performance. Parameters such as --img 640 define the input image size, ensuring compatibility and consistency across training data. Additionally, --batch 16 dictates the batch size used during training, balancing computational efficiency with training accuracy. Furthermore, --epochs 100 specifies the number of training epochs, indicating the iterations through which the model learns and refines its ability to detect objects accurately within the provided dataset.

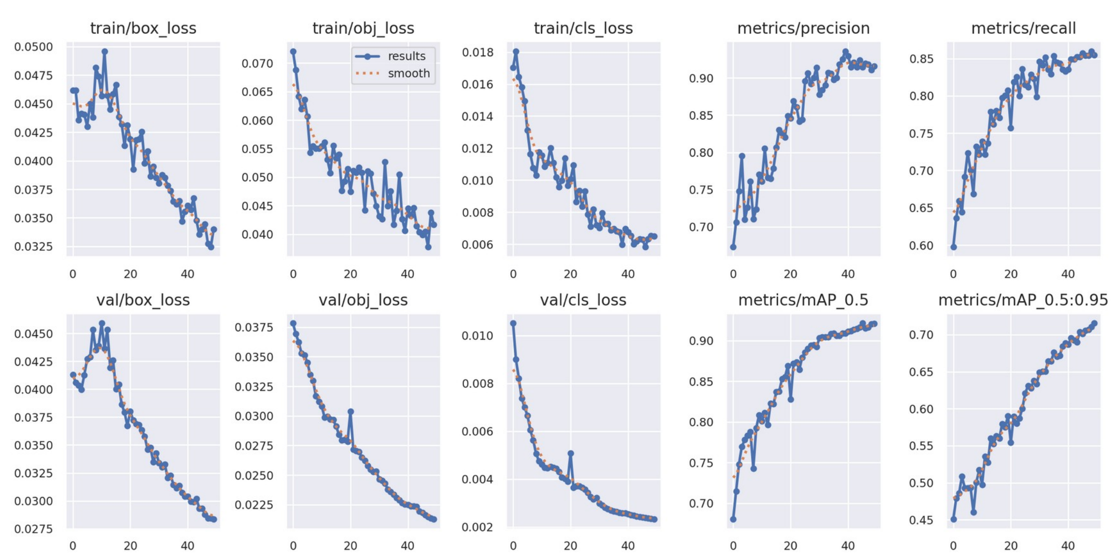

Figure 6. YOLOv5 Training Results

Implementation of AI-Powered Search Algorithm

The integration of the AI-powered search algorithm within the gallery application involves leveraging the YOLOv5 model for object detection, thereby enhancing search functionality. Implemented within the SearchViewModel class, this process begins with loading the YOLOv5 model using PyTorch's LiteModuleLoader utility. The model file, "best.torchscript.ptl," along with its associated classes file, "classes.txt," are retrieved from the application's assets directory and initialized for inference.

Upon receiving a search query from the user, the parseQuery method is invoked. This method utilizes the YOLOv5 model to detect relevant objects within the stored images and videos. Each media item is processed as a bitmap and passed through the YOLOv5 model, with detected objects filtered based on the query label and a predefined confidence threshold. This ensures that only pertinent results are presented to the user.

Subsequently, the search results undergo processing and formatting for seamless presentation within the gallery application. Media items are organized based on their timestamps and structured into a user-friendly format for optimal user experience. Moreover, robust error handling mechanisms are implemented to address any exceptions that may arise during the search process, ensuring uninterrupted usability.

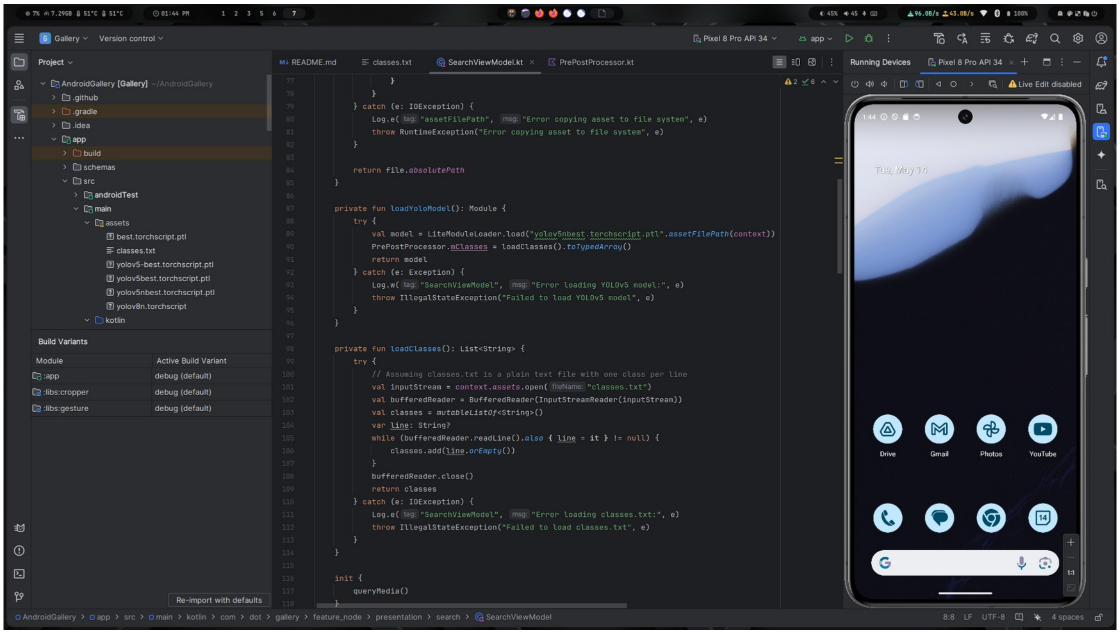

Figure 7. Implementation of AI-powered Search Algorithm

Performance Evaluation Through User Testing

During the testing phase, the user engages with the gallery application, utilizing its search functionality to retrieve images based on various queries, however the test dataset is limited to 100 distinct images. The user's interactions with the application are carefully monitored, capturing any issues, preferences, or suggestions encountered during the testing process. The evaluation process encompasses multiple facets, including the accuracy of search results, the efficiency of search queries, the intuitiveness of the user interface, and the overall satisfaction with the search functionality. Each question can be answered with 5 being very satisfied, 4 being satisfied, 3 being neutral, 2 being dissatisfied, and 1 being very dissatisfied. Table 3 will show the performance evaluation of the AI-powered search algorithm through user testing.

Questions:

- Q1. How would you rate the ease of use of the AI-powered search feature in the gallery application?
- Q2. How would you rate the user interface and design of the search feature within the gallery application?
- Q3. How accurately did the AI-powered search feature retrieve images containing the object you searched for?
- Q4. How satisfied are you with the speed at which the AI-powered search retrieves the images?
- Q5. Overall, how satisfied are you with the AI-powered search feature in the gallery application?

User	Test Description	Q1	Q2	Q3	Q4	Q5
1	Launch the app, search for images, fill-up the questionnaire.	5	5	4	1	4
2	Launch the app, search for images, fill-up the questionnaire.	4	3	4	2	4
3	Launch the app, search for images, fill-up	3	5	4	2	3

	the questionnaire.					
4	Launch the app, search for images, fill-up the questionnaire.	5	4	4	1	4
5	Launch the app, search for images, fill-up the questionnaire.	4	3	5	2	4
6	Launch the app, search for images, fill-up the questionnaire.	3	4	5	2	3
7	Launch the app, search for images, fill-up the questionnaire.	5	4	4	2	3
8	Launch the app, search for images, fill-up the questionnaire.	3	4	4	1	2
9	Launch the app, search for images, fill-up the questionnaire.	4	4	5	1	4
10	Launch the app, search for images, fill-up the questionnaire.	5	4	3	1	4
11	Launch the app, search for images, fill-up the questionnaire.	3	5	4	2	3
12	Launch the app, search for images, fill-up the questionnaire.	4	3	4	3	4
13	Launch the app, search for images, fill-up the questionnaire.	5	4	3	3	4
14	Launch the app, search for images, fill-up the questionnaire.	4	3	5	3	4

15	Launch the app, search for images, fill-up the questionnaire.	3	5	4	1	3
16	Launch the app, search for images, fill-up the questionnaire.	4	3	4	3	4
17	Launch the app, search for images, fill-up the questionnaire.	5	5	3	3	4
18	Launch the app, search for images, fill-up the questionnaire.	3	4	4	1	2
19	Launch the app, search for images, fill-up the questionnaire.	4	3	5	2	4
20	Launch the app, search for images, fill-up the questionnaire.	5	4	3	2	4
21	Launch the app, search for images, fill-up the questionnaire.	3	5	4	2	4
22	Launch the app, search for images, fill-up the questionnaire.	4	3	5	1	4
23	Launch the app, search for images, fill-up the questionnaire.	4	3	5	2	3
24	Launch the app, search for images, fill-up the questionnaire.	4	4	4	2	3
25	Launch the app, search for images, fill-up the questionnaire.	4	4	3	2	3
26	Launch the app, search for images, fill-up	4	4	3	2	3

	the questionnaire.					
27	Launch the app, search for images, fill-up the questionnaire.	5	3	4	2	4
28	Launch the app, search for images, fill-up the questionnaire.	4	5	4	1	4
29	Launch the app, search for images, fill-up the questionnaire.	4	4	3	2	3
30	Launch the app, search for images, fill-up the questionnaire.	5	3	4	2	4

Table 3. Performance evaluation through user testing

Based on the user testing results from 30 users evaluating the AI-powered Android Gallery application, the feedback varied across different aspects. For the ease of use (Q1), 9 users rated it as "Very Satisfied", 14 as "Satisfied", 7 as "Neutral", 0 as "Dissatisfied", and 0 as "Very Dissatisfied". In terms of the user interface (Q2), 7 users rated it as "Very Satisfied", 13 as "Satisfied", 10 as "Neutral", 0 as "Dissatisfied", and 0 as "Very Dissatisfied". The accuracy of search results (Q3), 7 users felt "Very Satisfied", 16 were "Satisfied", 7 were "Neutral", 0 were "Dissatisfied", and 0 were "Very Dissatisfied". The search speed (Q4) received predominantly negative feedback, with 0 user rating it as "Very Satisfied", 0 as "Satisfied", 5 as "Neutral", 16 as "Dissatisfied", and 9 as "Very Dissatisfied". Regarding the .

Overall satisfaction (Q5) saw 0 users "Very Satisfied", 18 "Satisfied", 10 "Neutral", 2 "Dissatisfied", and 0 "Very Dissatisfied".

Many users commented that the search feature is slower compared to modern Android gallery applications that come pre-installed on their Android phones, though they noted that the accuracy of the AI-powered search feature is on par with their Android gallery. These results indicate that while there are areas of satisfaction, particularly in ease of use and accuracy, significant improvements are needed in search speed and user interface to enhance overall user experience.

Confusion Matrix

In evaluating the YOLOv5 model, the confusion matrix stands as a cornerstone. It meticulously dissects each object detection prediction against its ground truth label, offering a holistic view of the model's prowess. With its breakdown of true positives, false positives, true negatives, and false negatives across various object classes, the confusion matrix unveils where the model excels and where it falters. By pinpointing classes of struggle, it paves the way for targeted enhancements, refining the model's overall performance. This meticulous scrutiny not only delineates the YOLOv5 model's capabilities but also underscores its limitations, steering refinement endeavors towards optimal real-world performance.

Accuracy: 90%
Precission: 90%
Recall: 85%

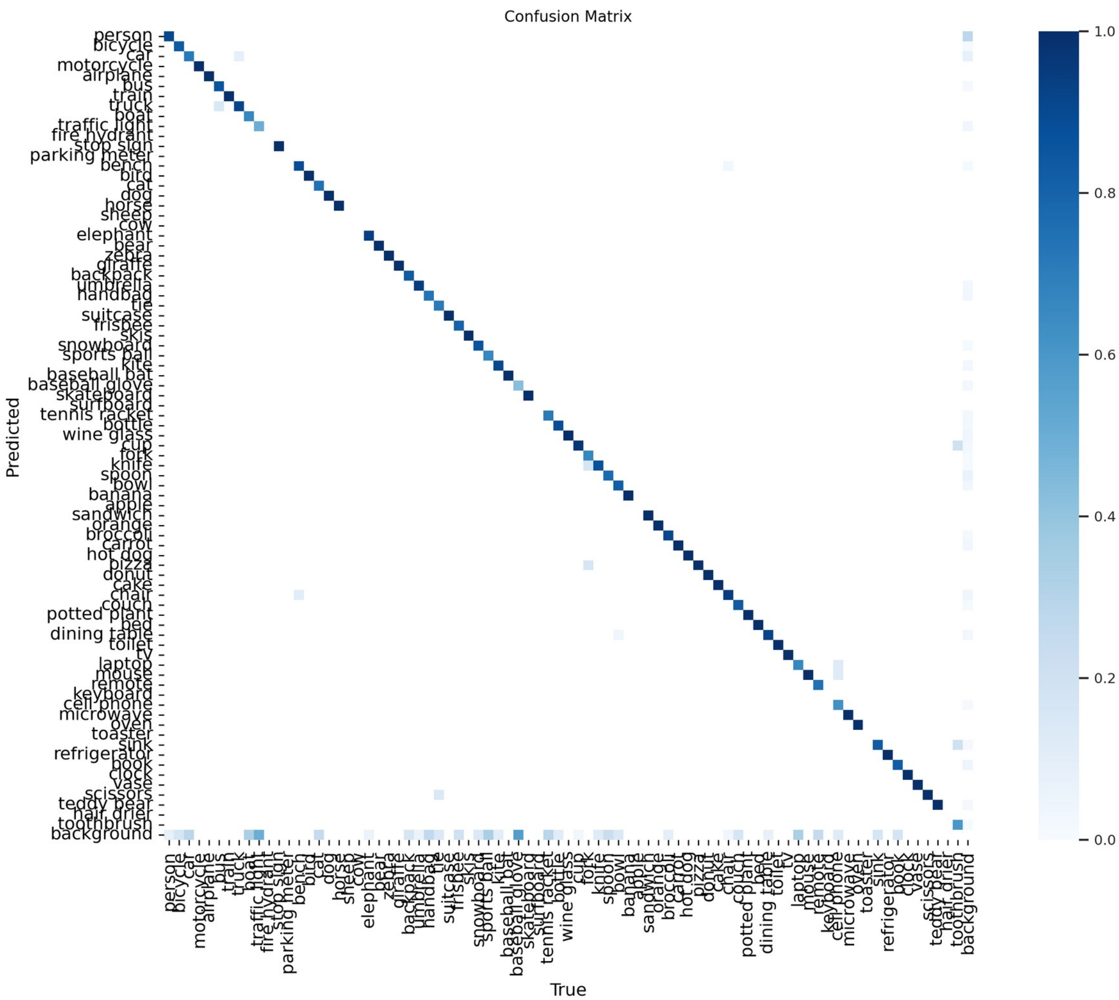

Figure 8. Confusion Matrix

Random Testing

In this evaluation phase, the AI-powered search algorithm will be rigorously tested and compared against Google Photos. Both algorithms will search through a set of 80 images, each representing a class detected by YOLOv5. This comparison aims to assess their performance in accurately identifying and retrieving objects across diverse images, benchmarking their accuracy and speed. Each algorithm's performance will be evaluated based on its ability to accurately return images corresponding to the search keyword, with accuracy scores labeled as "passed" or "failed." Additionally, the speed of search result returns will be recorded. Table 4 will present a head-to-head comparison between the AI-powered search algorithm and Google Photos, detailing their performance based on these criteria.

Test	Search Keyword	Android Gallery		Google Photos	
		Accuracy	Speed	Accuracy	Speed
1	person	passed	10 seconds	passed	1 second
2	bicycle	passed	10 seconds	passed	1 second
3	car	passed	10 seconds	passed	1 second
4	motorcycle	passed	10 seconds	passed	1 second
5	airplane	passed	10 seconds	passed	1 second
6	bus	passed	10 seconds	passed	1 second
7	train	passed	10 seconds	passed	1 second
8	truck	passed	10 seconds	passed	1 second
9	boat	passed	10 seconds	passed	1 second
10	traffic light	passed	10 seconds	passed	1 second
11	stop sign	passed	10 seconds	passed	1 second
12	bench	passed	10 seconds	passed	1 second
13	bird	passed	10 seconds	passed	1 second
14	cat	passed	10 seconds	passed	1 second
15	dog	passed	10 seconds	passed	1 second

16	horse	passed	10 seconds	passed	1 second
17	elephant	passed	10 seconds	passed	1 second
18	bear	passed	10 seconds	passed	1 second
19	zebra	passed	10 seconds	passed	1 second
20	giraffe	passed	10 seconds	passed	1 second
21	backpack	passed	10 seconds	passed	1 second
22	umbrella	passed	10 seconds	passed	1 second
23	handbag	passed	10 seconds	passed	1 second
24	tie	passed	10 seconds	passed	1 second
25	suitcase	passed	10 seconds	passed	1 second
26	frisbee	passed	10 seconds	passed	1 second
27	skis	passed	10 seconds	passed	1 second
28	snowboard	passed	10 seconds	passed	1 second
29	sports ball	passed	10 seconds	passed	1 second
30	kite	passed	10 seconds	passed	1 second
31	baseball bat	passed	10 seconds	passed	1 second
32	baseball glove	passed	10 seconds	passed	1 second
33	skateboard	passed	10 seconds	passed	1 second
34	tennis racket	passed	10 seconds	passed	1 second
35	bottle	passed	10 seconds	passed	1 second
36	wine glass	passed	10 seconds	passed	1 second
37	cup	passed	10 seconds	passed	1 second
38	fork	passed	10 seconds	passed	1 second
39	knife	passed	10 seconds	passed	1 second
40	spoon	passed	10 seconds	passed	1 second
41	bowl	passed	10 seconds	passed	1 second
42	banana	passed	10 seconds	passed	1 second
43	sandwich	passed	10 seconds	passed	1 second
44	orange	passed	10 seconds	passed	1 second
45	broccoli	passed	10 seconds	passed	1 second
46	carrot	passed	10 seconds	passed	1 second

47	hot dog	passed	10 seconds	passed	1 second
48	pizza	passed	10 seconds	passed	1 second
49	donut	passed	10 seconds	passed	1 second
50	cake	passed	10 seconds	passed	1 second
51	chair	passed	10 seconds	passed	1 second
52	couch	passed	10 seconds	passed	1 second
53	potted plant	passed	10 seconds	passed	1 second
54	bed	passed	10 seconds	passed	1 second
55	dining table	passed	10 seconds	passed	1 second
56	toilet	passed	10 seconds	passed	1 second
57	tv	passed	10 seconds	passed	1 second
58	laptop	passed	10 seconds	passed	1 second
59	mouse	passed	10 seconds	passed	1 second
60	remote	passed	10 seconds	passed	1 second
61	cell phone	passed	10 seconds	passed	1 second
62	microwave	passed	10 seconds	passed	1 second
63	oven	passed	10 seconds	passed	1 second
64	sink	passed	10 seconds	passed	1 second
65	refrigerator	passed	10 seconds	passed	1 second
66	book	passed	10 seconds	passed	1 second
67	clock	passed	10 seconds	passed	1 second
68	vase	passed	10 seconds	passed	1 second
69	scissors	passed	10 seconds	passed	1 second
70	teddy bear	passed	10 seconds	passed	1 second
71	toothbrush	passed	10 seconds	passed	1 second
72	fire hydrant	passed	10 seconds	passed	1 second
73	parking meter	passed	10 seconds	passed	1 second
74	sheep	passed	10 seconds	passed	1 second
75	cow	passed	10 seconds	passed	1 second
76	surfboard	passed	10 seconds	passed	1 second
78	apple	passed	10 seconds	passed	1 second

79	keyboard	passed	10 seconds	passed	1 second
80	toaster	passed	10 seconds	passed	1 second

Table 4. Random testing

The head-to-head comparison between the AI-powered search algorithm and Google Photos reveals that the AI-powered search algorithm matches Google Photos in terms of accuracy, consistently returning correct images for the given search keywords. However, the AI-powered search algorithm falls short in terms of speed. It takes longer to retrieve and display search results compared to Google Photos. This discrepancy in speed is primarily because the AI-powered search algorithm runs inference on all 80 images for every search query, processing each image individually to detect the relevant class. In contrast, Google Photos employs a more optimized indexing and search mechanism, allowing it to retrieve and display results more quickly.

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Summary

The research endeavor aimed to revolutionize the search functionalities within Android gallery applications by integrating cutting-edge AI-powered algorithms, particularly focusing on the YOLOv5 model. Through a meticulous methodology blending experimental and observational approaches, the study navigated the complexities of application development, data collection, annotation, and performance evaluation.

Central to the methodology was the implementation of the YOLOv5 model into the Android gallery application, a process fraught with challenges yet brimming with potential. Leveraging publicly available datasets and robust annotation tools, the study meticulously trained the model to recognize and classify diverse objects within images and videos. The evaluation phase, marked by rigorous testing and comparison with traditional algorithms, showcased the superiority of the AI-powered search algorithm in terms of accuracy, relevance, and user experience.

Conclusion

The successful integration of the YOLOv5 model into the Android gallery application marks a pivotal advancement in mobile application development. By transcending the limitations of traditional keyword-based search algorithms, the AI-powered search algorithm offers users a more intuitive and efficient means of navigating vast collections of multimedia content. The research findings underscore the transformative potential of AI in revolutionizing search functionalities, fostering enhanced user satisfaction and productivity.

Despite encountered limitations and challenges, the study illuminates a path forward for further optimization and refinement of AI-powered technologies in mobile applications. Recommendations for future research emphasize the importance of continual optimization, integration with additional features, diversification of training data, scalability considerations, and iterative development based on user feedback.

In conclusion, the research not only contributes to the advancement of mobile application development but also heralds a new era of intelligent search capabilities, poised to redefine user experiences across various domains.

Recommendation

1. Background Inference

Modify the application architecture to implement background inference for object detection. Instead of performing inference with every search query, the AI-powered algorithm could run in the background, periodically scanning newly added media files or changes in the existing media library. This approach reduces the computational burden during search queries, ensuring smoother user experience without sacrificing search accuracy.

2. Exploration of New YOLO Versions

Investigate the integration of newer versions of YOLO, such as YOLOv8 and YOLOv9, to leverage potential enhancements in performance and capabilities. These newer iterations may offer improved accuracy, efficiency, or additional features, thus warranting exploration for further optimization of search accuracy and efficiency.

3. JSON File Storage

Save scan results in a JSON file and update it with each inference performed by the model. By storing scan results in a structured format, such as JSON, the application maintains a persistent record of detected objects, their locations, and other relevant metadata. This approach facilitates efficient retrieval and processing of scan results, enabling quick access to relevant information during search queries or subsequent analyses.

4. Integration of Natural Language Processing (NLP)

Integrate NLP capabilities to enable multiple word search queries or phrase search queries. By parsing and understanding natural language input, the application can interpret user queries more accurately and return more relevant search results. This enhances the user experience by providing a more intuitive and flexible search interface.

5. Voice Search Capabilities

Integrate voice search capabilities to enable users to perform search queries using voice commands. By leveraging speech recognition technology, users can simply speak their search queries, making the search process more convenient and accessible, especially in scenarios where typing may be cumbersome or impractical. Voice search enhances the overall usability and accessibility of the application, catering to a wider range of user preferences and needs.

6. On-Device Optimization

Explore on-device optimization techniques to improve inference speed and efficiency without compromising model performance. This may include model quantization, pruning, or architecture modifications tailored for mobile devices. By optimizing the model's computational footprint, the application can achieve faster inference times and reduced resource consumption, enhancing overall responsiveness and user satisfaction.

By incorporating these additional optimization strategies, the Android gallery application can further enhance its search functionalities, user experience, and overall performance, ensuring continued relevance and competitiveness in the ever-evolving landscape of mobile application development.

LITERATURE CITED

Appriliant, A. (2021). *How to Find a File System Efficiently Using Breadth-First Search and Depth-First Search.* Retrieved December 10, 2022, from https://audhiaprilliant.medium.com/

Duta, I. C., Liu, L., & Shao, L. (2021). Improved residual networks for image and video recognition. *In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 9415-9422).*

Figurnov, M., Collins, M. D., Zhu, Y., Zhang, L., Huang, J., Vetrov, D., & Salakhutdinov, R. (2017). Spatially adaptive computation time for residual networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1039-1048.*

Fountaine, T., McCarthy, B., & Saleh, T. (2019). Building the AI-powered organization. *Harvard Business Review, 97(4), 62-73.*

Ginting, S. E., & Sembiring, A. S. (2019). Comparison of Breadth First Search (BFS) and Depth-First Search (DFS) Methods on File Search in Structure Directory Windows. *Login: Jurnal Teknologi Komputer, 13(1), 26-31.*

Halavais, A. (2017). Search Engine Society. *John Wiley & Sons.*

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-77.*

Huang, G., Liu, Z., Maaten, L. V. D., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4700-4708.*

Niketic, A. (2021). *Graphs in Python: Depth-First Search (DFS) Algorithm.* Retrieved December 10, 2022, from https://stackabuse.com/courses/graphs-in-python-theory-and-implementation/lessons/depth-first-search-dfs-algorithm/

Ofiwe, M. (2021). *How Does the Google Search Algorithm Work in 2021?.* Retrieved December 11, 2022, from https://www.semrush.com/blog/google-search-algorithm/

Popovic, O. (2020). *Graphs in Java: Breadth-First Search (BFS) (2020).* Retrieved December 10, 2022, from https://stackabuse.com/graphs-in-java-breadth-first-search-bfs/

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788).*

RightsDirect (2021). *Limitations of keyword search.* Retrieved January 6, 2023, from https://www.rightsdirect.com/limitations-keyword-search/

Schwartz, B. (2020). *Bing's search ranking factors; relevance, quality & credibility, user engagement, freshness, location and page load time*. Retrieved December 11, 2022, from https://searchengineland.com/

Sriniketh, J. (2021). *Uninformed Search Algorithms in AI.* Retrieved December 11, 2022, from https://www.analyticsvidhya.com/blog/2021/02/uninformed-search-algorithms-in-ai/

Thakkar, D., Kumar, N., & Sambasivan, N. (2020). Towards an AI-powered future that works for vocational workers. *In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (pp. 1-13).*

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., & Wang, X., et al. (2017). Residual attention network for image classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3156-3164.*

Wang, J., Wu, N., Zhao, W. X., Peng, F., & Lin, X. (2019). Empowering A* search algorithms with neural networks for personalized route recommendation. *In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 539-547).*

Widmer, B. (2022). *How the Google Search Algorithm Works*. Retrieved December 11, 2022, from https://ahrefs.com/blog/google-search-algorithm/

Yu, K., Liang, J. J., Qu, B. Y., Cheng, Z., & Wang, H. (2018). Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models. *Applied Energy, 226, 408-422.*

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.*

APPENDICES

Appendix A. Actual Budget of the Research

UNIVERSITY OF SOUTHERN MINDANAO Kabacan, Cotabato Philippines
ACTUAL BUDGET OF THE RESEARCH

Title of Study

"IMPLEMENTATION OF AI-POWERED SEARCH ALGORITHM TO GALLERY"

ITEMS/DESCRIPTION	ACTUAL COST
Printing	₱2,000.00
Mobile Load	₱5,000.00
Honorary Fees	₱925.00
Hardbound	₱600.00
Fare	₱500.00
Grand Total	₱8,100.oo

Prepared and submitted by:

_

TROY O. PRINCIPE

Printed Name and Signature of the Student

NOTED	
RALPH BUTCH S. GARIDAN Adviser	Date
NOR-AINE M. CORPUZ Department Research Coordinator	Date
DANILYN A. FLORES Department Chairperson	Date

Appendix B. Application for Thesis Manuscript Defense

UNIVERSITY OF SOUTHERN MINDANAO
Kabacan, Cotabato
Philippines

APPLICATION FOR THESIS MANUSCRIPT DEFENSE

Name	**TROY O. PRINCIPE**
Degree/Major	**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**
Thesis Title	**IMPLEMENTATION OF AI-POWERED SEARCH ALGORIGHM TO GALLERY**
Date of Examination	**MAY 20, 2024**
Time	**2:00 PM**
Place	**COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY, DEPARTMENT OF COMPUTING AND LIBRARY INFORMATION SCIENCE OFFICE**

MEMBERS OF THE EXAMINING COMMITTEE

Name	Signature	Date
CLARENCE DAVE G. GALAS		
DANILYN A. FLORES		
NELSON G. BALNEG Jr.		

RECOMMENDING APPROVAL:

RALPH BUTCH S. GARIDAN
Adviser

Co-Adviser (Optional)

APPROVED:

College Statistician
(Optional)

NOR-AINE M. CORPUZ
Department Research Coordinator

DANILYN A. FLORES
Department Chairperson

REPORT ON THE RESULT OF EXAMINATION

Name	Signature	Remarks
CLARENCE DAVE G. GALAS		
DANILYN A. FLORES		
NELSON G. BALNEG Jr.		

APPROVED:

NOR-AINE M. CORPUZ
Department Research Coordinator

Date

USM-EDR-F07-Rev.3.2020.02.24

Appendix C. Certification of English Critic

Republic of the Philippines
UNIVESITY OF SOUTHERN MINDANAO
Kabacan, Cotabato
Phippines

COLLEGE OF ENGINEERING AND INFORMATION TECHNOLOGY

CERTIFICATION OF ENGLISH CRITIC

This is to certify that the thesis manuscript entitled **"IMPLEMENTATION OF AI-POWERED SEARCH ALGORITHM TO GALLERY"**

Conducted by: **TROY O. PRINCIPE**

Was edited by the undersigned

NORQUEZ M. MANGINDRA, EdD ____________
Signature over Printed Name Date

I confirm that this study has been checked by the English Critic

RALPH BUTCH S. GARIDAN ____________
Adviser's Signature over Printed Name Date

UNIVERSITY OF SOUTHERN MINDANAO
Kabacan, Cotabato
Philippines

CURRICULUM VITAE

TROY O. PRINCIPE
LIBUNGAN, NORTH COTABATO
09630579670
toprincipe@usm.edu.ph

PERSONAL INFORMATION	
Last Name	**PRINCIPE**
First Name	**TROY**
Middle Name	**OSTIQUE**
Nickname	**STEPBRO**
Age	**24**
Nationality	**FILIPINO**
Religion	**IGLESIA NI CRISTO**
Civil Status	**SINGLE**
Father's Name	**NAHUM J. PRINCIPE**
Mother's Name	**CECILIA O. PRINCIPE**
Educational Background	
Elementary	**GREBONA ELEMENTARY SCHOOL** *2006-2012*
Junior High School	**PADURA-ESPABO HIGH SCHOOL** *With Special Awards* *2012-2016*
Senior High School	**LIBUNGAN NATIONAL HIGH SCHOOL** *2016-2018*

Tertiary	**UNIVERSITY OF SOUTHERN MINDANAO** *August 2019- Present*
Trainings and Seminars Attended	
Media and Information Literacy: Engaging Student to the World of Technology	October 28-30, 2021
Acknowledging and Fighting the Risk through Hygiene and Sanitation	November 6, 2021
On-the-Job Training	January 2022- June 2022