QUALITY ANALYSIS OF SHOPEE SELLER PORTAL AND SEMI-SUPERVISED LEARNING APPROACH FOR AUTOMATIC DETECTION AND FASHION PRODUCT CATEGORY PREDICTION USING FC-YOLOV4

YAMIN THWE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF
SCIENCE PROGRAM IN DATA AND INFORMATION SCIENCE
FACULTY OF SCIENCE AND TECHNOLOGY
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI
ACADEMIC YEAR 2022

QUALITY ANALYSIS OF SHOPEE SELLER PORTAL AND SEMI-SUPERVISED LEARNING APPROACH FOR AUTOMATIC DETECTION AND FASHION PRODUCT CATEGORY PREDICTION USING FC-YOLOV4

YAMIN THWE

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF
SCIENCE PROGRAM IN DATA AND INFORMATION SCIENCE
FACULTY OF SCIENCE AND TECHNOLOGY
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI
ACADEMIC YEAR 2022**

Thesis Title	Quality Analysis of Shopee Seller Portal and Semi-Supervised Learning Approach for Automatic Detection and Fashion Product Category Prediction Using FC-YOLOv4
Name-Surname	Miss Yamin Thwe
Program	Data and Information Science
Dissertation Advisor	Assistant Professor Nipat Jongsawat, Ph.D.
Dissertation Co-Advisor	Assistant Professor Anucha Tungkasthan, Ph.D.
Academic Years	2022

THESIS COMMITTEE

.... Chairman
(Mr. Parham Porauhan, Ph.D.)

.... Committee
(Assistant Professor Pitaya Poompuang, Ph.D.)

.... Committee
(Assistant Professor Anucha Tungkasthan, Ph.D.)

Nipat Jongsawat
.... Committee
(Assistant Professor Nipat Jongsawat, Ph.D.)

Approved by the Faculty of Science and Technology, Rajamangala University of Technology Thanyaburi in Partial Fulfillment of the Requirement for the Master's degree

Nipat Jongsawat
.............................Dean of Faculty of Science and Technology
(Assistant Professor Nipat Jongsawat, Ph.D.)
Date 28 Month December Year 2022

Thesis Title	Quality Analysis of Shopee Seller Portal and Semi-Supervised Learning Approach for Automatic Detection and Fashion Product Category Prediction Using FC-YOLOv4
Name-Surname	Miss Yamin Thwe
Program	Data and Information Science
Dissertation Advisor	Assistant Professor Nipat Jongsawat, Ph.D.
Dissertation Co-Advisor	Assistant Professor Anucha Tungkasthan, Ph.D.
Academic Years	2022

ABSTRACT

Consumers have shifted away from traditional transactions to online shopping as a result of the COVID-19 pandemic, digital social life, and staying at home. This significantly impacts the volume of transactions for several e-commerce platforms, including Shopee, which must continue improving customer satisfaction. One way to accomplish this is to improve the accuracy of the automatic category recommendation, which currently contains a significant number of errors made by the seller. Over the past few decades, object identification in images has evolved quickly. Achieving fast and accurate detection of fashion products in the e-commerce environment is essential for selecting the right category. Nowadays, e-commerce sites provide the purchase of both new and second-hand clothing. Therefore, when categorizing fashion clothing, it is essential to be able to categorize it precisely, regardless of the cluttered background.

The Shopee Data Scraper was used to collect data, and two qualitative analyses, namely content and thematic analysis were used to determine how many instances of automatic category recommendation fraud were committed by the seller. This research also proposes the Fashion Category detector FC-YOLOv4 algorithm for the detection of multi-class fashion products and accessories categories. We present recently acquired tiny product images with various resolutions, sizes, and position datasets from the Shopee E-commerce (Thailand) website. We used the semi-supervised learning approach to reduce image labeling time, and the number of resulting images is then increased through the use of brightness, mosaic, rotation, and CLAHE augmentation.

According to the content analysis, there is a 29 percent error in product category selection. Meanwhile, 75.1 percent of products should be classified separately but are instead classified as 'others.' Following that, 72.7 percent of product titles contain the same words as existing categories. From this research, we analyzed the percentage of errors in the automatic category recommendation mechanism from the Shopee platform, which causes sellers to place their products in the wrong category so that they can be used as suggestions for improvement or further advance of research. And our FC-YOLOv4 approach results in reasonable Average Precision (AP), Mean Average Precision (mAP), True or False Positive (TP/ FP), Recall, Intersection over Union (IoU), and reliable object detection. According to experimental findings, our model increases the mAP by 0.07 percent and 40.2 percent increment compared to the original YOLOv4 and YOLOv3. Experimental findings from our FC-YOLOv4 model demonstrate that it can effectively provide accurate fashion category detection for properly captured and clutter images compared to the YOLOv4 and YOLOv3 models. Then, we utilized TensorFlow Lite to implement our FC-YOLOv4 model on the android platform. Then, we compared the detection findings of our FC-YOLOv4 Android software with those of the Shopee Thailand Seller Web Portal.

Keywords: content analysis, thematic analysis, product categorization, semi-supervised learning, YOLOv4

ACKNOWLEDGMENT

This research would not have been possible without the contributions of numerous individuals. My sincere appreciation goes to my advisor, Assistant Professor Nipat Jongsawat, who read my countless edits and helped me make sense of the confusion. My gratitude also goes to my co-advisor, Assistant Professor Anucha Tungkasthan, who provided direction and support. I cannot adequately express my appreciation to my adviser and co-advisor for their tremendous patience and suggestions. I also could not have embarked on this road without the information and advice of my defense committee.

Thanks to the Rajamangala University of Technology Thanyaburi for the collaboration, support, and computation resources for our experiments. Also, appreciate the E-CUBE I scholarship program for giving me the chance to continue this research.

Lastly, I would be remiss in not mentioning my family and friends. Their belief in me has strengthened my spirit and motivation throughout this endeavor.

Yamin Thwe

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The purpose of the thesis is to determine the percentage of mistakes in the automatic category suggestion mechanism that results in sellers placing their items in the incorrect category, particularly in clothes fashion, and accessories, and to detect and classify such types automatically in order to make it easy, accurate, and quick for sellers to submit their product, as well as to improve Shopee's seller recommendation system. In this chapter, we discussed the background and statement of the problems, purpose of the study, research questions and hypothesis, research procedure, limitation and significance of our study, and outline.

1.1 Background and Statement of the Problems

The pandemic of Covid-19 has accelerated the development of E-commerce. Many buyers have been compelled to make purchases online due to the pandemic, as many retail outlets worldwide are either closed according to lockdown measures or have little capacity to maintain social separation. Most analysts and e-commerce companies believe that online purchasing will continue to rise gradually even after the pandemic ends. As a result, specialists are doing comprehensive studies to improve the quality of e-commerce and keep existing clients while attracting new ones.

Through their research, the more relevant information regarding a client's demand we can present during the search and suggestion process, the greater the level of customer satisfaction. Customer happiness is crucial to an e-commerce platform's success (Choshin & Ghaffari, 2017). Customer satisfaction is highly related to identifying products that meet each consumer's individual requirements. Prior to the pandemic, one of the keys to the success of e-commerce platforms was customer review ratings, but not after the outbreak (Agus, Yudoko, Mulyono, & Imaniya, 2021). We learned that product categorization plays a critical role in e-commerce and can directly impact other services such as product recommendation and search.

In the early days of E-commerce, product categorization was done manually, as integrating thousands of products daily would be prohibitively expensive and time-consuming. Recent advances in machine learning (ML) have also been incorporated into innovative business applications and e-commerce administrations to enable them to reason about complex systems and provide better solutions. Image processing and computer vision heavily depend on object detection ability(Arulprakash & Aruldoss, 2021). Computer vision is a widely utilized technique in a variety of applications, including clothing detection (Z. Liu, Luo, Qiu, Wang, & Tang, 2016; Wu et al., 2021), clothes collocation [8]– [10], clothing attribution and category recognition (W. Wang, Xu, Shen, & Zhu, n.d.). Clothing is an inescapable requirement for improving personal looks in people's lives. This, of course, affects how vendors meet customer needs on e-commerce platforms.

The diversity of the internet trade industry development allows users to determine which sites profited the most. Among the popular e-commerce platforms, Shopee Pte Ltd is one of the online marketplaces that link sellers and buyers of various products. Shopee is a multinational technology business headquartered in Singapore that specializes in e-commerce. The Shopee Seller Portal assists sellers by enabling them to simply register, administer, and promote their products and manage payment choices. To upload a product, the seller must complete the following fields: product name, product description, category, brand attribute, and additional attributes. Vendors can easily select their product's category, sub-category, and third-level category. Additionally, the system proposes the type that is most closely related to the phrase. Even though the system suggests a proper category for the products on the Shopee seller portal, sellers frequently make category mistakes, particularly with fashion products and accessories. Incorrect category selection can result in deceptive customer searches and recommendations, eroding customer trust.

We choose the object of Shopee online buying and selling platform because it is among one of the leading marketplaces in the Southeast Asian region and has been a preferred shopping destination for the Southeast Asian population (David Tinambunan, 2019). They provide various kinds of necessities ranging from home needs, office needs, and baby needs to adult needs. The objective is to provide consumers with a list of desired products when they conduct a product search and to enhance the percentage of product purchases. In this study, we examined how many automatic category recommendation mechanisms included in the Shopee seller portal can help prevent seller fraud and determine which categories most vendors choose incorrectly. And we'll propose a new object detection model called Fashion Category Detector FC-YOLOv4 object identification technology to recommend the product category automatically, supporting merchants in choosing the optimal pick. This is a necessary component of e-commerce business intelligence.

1.2 Purpose of the Study

The purpose of this research is to identify and classify clothing fashion and accessories. The main research contents include:

1. Propose an analysis to compare and verify the correctness of a seller's picture and category selection on the Shopee website.
2. Propose a recently acquired, tiny dataset from the Shopee E-commerce (Thailand) website with a total of 30,483 photographs.
3. Propose an approach for automatically annotating image data to reduce manual labelling time and increase object detection accuracy.
4. Propose an accurate image-based classification model based on FC-YOLOv4 capable of classifying Shopee Thailand product images into categories and identifying their types.
5. Propose the Android deployment strategy with FC-YOLOv4 and the open-source TensorFlow Lite framework.

1.3 Research Questions and Hypothesis

Several experiments will be carried out to answer the following questions:

1. How to assemble the seller's most misleading categories?
2. How to decrease the amount of time necessary for hand annotation of images used in object detection?

3. How to configure the seller portal's category recommendation system to select a product category automatically based on product images?
4. How can the proposed Fashion categories prediction FC-YOLOv4 model be implemented in mobile devices?

1.4 Research Procedures

The following procedures are used to categorize apparel goods in this study.

1. We apply the Shopee Data Scraper (Data Sunday) tool to extract product data from the Women's Clothes category on the Shopee website (www.shopee.co.th).
2. We apply Content Analysis and Thematic Analysis to determine the seller's most erroneous category selections.
3. We acquire a new product image dataset from Shopee and Google Image for fashion category classification.
4. We apply LabelImg to do image data labelling.
5. We apply the semi-supervised learning approach for automatically annotating data to save manual tasks and increase the number of images by using image augmentation to improve object detection accuracy.
6. We apply image data augmentation technology to expand the quantity and quality of the dataset.
7. We enhance the number of short-circuiting and stacking to improve network performance and features.
8. We apply an accurate image-based classification model utilizing FC-YOLOv4 that can classify Shopee Thailand product images and identify their types.
9. We employ the TensorFlow Lite framework for the FC-YOLOv4 model weights that have been pre-trained in order to create the category recommendation system in the mobile application system.
10. We tested the models with unseen clutter background images. We compared our FC-YOLOv4 model to the YOLOv4 and YOLOv3 models in terms of detection time and accuracy to verify the applicability and high efficiency

of the proposed method in fashion category detection. We analyze the suggested FC-YOLOv4 model on mobile devices and compare its results to those of the Shopee Seller Portal Recommendation System to determine its efficacy.

1.5 Limitations of the Study

Due to the fixed time frame, some limitations must be imposed on this research to ensure that it is completed on time and is more focused.

1. This work focuses exclusively on forecasting the product's category.
2. This work focuses on predicting only the primary category; it does not attempt to forecast the hierarchical category tree.
3. This work focuses exclusively on a dataset from Thailand's shopee.co.th for thematic and content analysis. Product categorization will be accomplished through the use of images from Shopee and Google Images.
4. This work focuses primarily on the category under “Women's Clothes” and “Fashion Accessories” main categories.

1.6 Significance of the Study

When sellers add a product to the Shopee website, the process of categorizing the goods is typically a laborious and error-prone procedure. This technique is particularly fascinating since it enables merchants to be indexed automatically by the system with the use of an object detection algorithm and image data. The FC-YOLOv4 model was used to classify product photos provided by vendors to Shopee's e-commerce platform. This model is chosen because its accuracy is superior to that of other models. The FC-YOLOv4 model that was trained on our proposed dataset was then deployed on mobile devices. The purpose of this study is to aid sellers in addressing the demands of their consumers and raising client demand.

1.7 Outline

The next chapter 2 will discuss the theoretical background of this study, with a particular emphasis on the discovery process and machine learning methodologies. Following a description of the methods used in Chapter 3, Chapter 4 will present empirical data and the corresponding experimental results. Chapter 5 will explain the findings, conduct a critical analysis of the methodologies and methods used, and assess the data's validity and reliability. Finally, chapter 6 will review the study and provide a perspective for further research and development on this subject.

CHAPTER 2

THEORIES AND LITERATURE REVIEW

This chapter discusses the theories and literature review associated with the thesis to explain the specifics of the research on predicting the category of products on the Shopee e-commerce platform using data analysis and machine learning process methods.

2.1 State of the Art

Data scientists and software engineers in large numbers are experimenting with various methodologies for categorizing e-commerce platforms automatically. This is to minimize the possibility of sellers making an inappropriate selection. Shopee's customer experience dimensions and satisfaction were analyzed in order to establish a link between the two in Bandung. Customer satisfaction is excellent at 69.1%. Shopee should be able to improve display arrangements for purchasing goods by enhancing the appearance of Shopee's needs so that customers can easily find the product they are looking for and increase customer comfort and satisfaction (Happ, Scholl-Grissemann, Peters, & Schnitzer, 2021). The impact of the business product category on consumer trust was researched regarding e-commerce in Malaysia. Apparel, computer and mobile gadgets, and electric and electronic equipment are the primary categories respondents placed their trust in when shopping online. 64.5 percent look for the best deal on selected items. And they provided an understanding of the consumer's belief in a product category before making a purchase (Mohd Dahlan, 2019).

There are two current e-commerce category issues: First, the hierarchy is shallow; frequently, there are too many products in a single category, making it difficult for a consumer to browse them. Second, because the hierarchy is created manually, it is not easy to update it as new products are introduced (Hsieh, Wu, Chen, & Yang, 2017). A product may be associated with several categories, with tens of thousands of existing classes. The Novel Neural Product Categorization Model (NPC) is designed to produce new product categories based on the contents of the products. When a consumer conducts a category search, they may click on the most suitable products inside that category. This indicates that frequently clicked products are more likely to fall under the category of inquiring (L. Tan, Li, & Kok, 2020).

Automated product classification in e-commerce using images greatly influences the retail business and is perhaps the most sophisticated application of computer vision. With the advancement of machine learning, several studies have been enhanced, and deep learning technology is now widely employed in the fashion and apparel industries. One of these studies is (Kim et al., 2021), a novel model appropriate for low-power devices that use a single-stage detector to recognize numerous garments in photos, utilizing the DeepFashion2 dataset, which has over 200,000 fashion photos annotated in 13 different classes. Compound scaling is used to scale a backbone feature network that trains input features at various resolutions. It is efficient due to its small number of parameters and cheap computing cost. They minimized the difficulties of single-stage detectors by utilizing the focused loss proposed by RetinaNet. They suggested a technique for multiple-clothing identification and estimating fashion landmarks as an adaption of EfficientDet. Without image preparation, the recommended approach is quick and accurate, and they obtained a bounding box detection accuracy of 0.686 mAP with an inference time of 42 milliseconds. It is extremely inefficient in terms of inference time and resource utilization. This could limit the application of fashion image analysis in real-life situations.

Furthermore, YOLOv4-TPD (YOLOv4 Two-Phase Identification) (Lee & Lin, 2021), a two-phase fashion garment recognition system based on the YOLOv4 algorithm, is also available. They utilized the Clothing Co-Parsing (CCP) dataset containing 2,098 high-resolution street photographs of fashion clothes. The model detection target categories included the jacket, the top, the bottoms, the skirt, and the bag. They suggested a two-phase transfer learning object detection model for detecting fashion apparel photographs with complicated backgrounds. The experimental results demonstrated that adopting two-phase transfer learning and the CLAHE image enhancement method can enhance model detection precision. They achieved the accuracy of 96.01% of mAP with 15.6333 milli-seconds detection time. This research is more relevant to others regarding detection time and precision. However, the CCP dataset only contains high-resolution photos with no overlapping or occluded items. The selected categories are also simple to identify due to their distinct patterns. This model is unsuitable for clothing with many styles captured by multiple suppliers.

A new system was proposed by incorporating the following properties into the standard item-based collaborative filtering method (K-RecSys) (Hwangbo, Kim, & Cha, 2018). The company's online and physical locations sell the same stuff. Second, fashion items are often seasonal, meaning that overall buyer preferences vary by season. Finally, the buyer usually acquires a product to replace a previously favored item or complement a previously purchased item. Another technique combining deep learning and image processing was used to detect and categorize Adidas AG(TM) apparel logos, stripes, colors, and other properties (Donati, Iotti, Mordonini, & Prati, 2019). Taking a previously trained learning model and retraining it on new data is referred to as fine-tuning. They utilized VGG19 to fine-tune the precision to its utmost value. As a consequence, they developed a revolutionary image recognition and feature extraction system that is very accurate at classifying images and is trustworthy and robust enough to be employed by a firm like Adidas.

When detecting fashion image categories, there are numerous categories that have similar patterns such as dress and skirt categories. Researchers are proposed in order to differentiate object-to-object detection in the previous research (Bossard et al., n.d.-a; H. Chen, Gallagher, & Girod, n.d.; N. Li, Cheng, & Zhang, 2022; Zhou, Qi, Wang, Shen, & Zhu, 2022; Zhou, Wang, Liu, Yang, & Gool, n.d.). To extract suitable and efficient adjacent objects in fashion detection,(M. Chen, Qin, Qi, & Sun, n.d.) suggested the Dual Attention Feature Enhancement (DAFE) module. Long-range modeling interactions between channels highlight task-related characteristics and improve pixel-level information. (Bossard et al., n.d.-b) suggested a method for detecting upper-body garments and classifying them according to distinct attributes for each observed garment. They developed 15 clothing classes for evaluation and presented an 80,000-image benchmark data set for the clothing classification job. Their classifier outperformed an SVM baseline on challenging benchmark data with 41.38 percent versus 35.07 percent average accuracy.

Moreover, to facilitate categorization (Oyewole & Olugbara, 2018) offered an upgraded color image-based product classification architecture for the e-commerce business that comprises a median filter, a newly built Eigen Colour feature (cHOG-ECF and cULBP-ECF), and an ensemble of artificial neural networks for product categorization. A novel technique for multi-class garment detection is proposed that adapts 3-level Spatial pyramid pooling (SPP), a multi-scale visual feature extractor, to enable faster and more accurate apparel recognition and classification utilizing a single-stage deep transfer learning model (SS-DTLM) (Kumar Addagarla, 2022). Additionally, this article highlighted the recognized image's distinct color spaces utilizing the SS-DTLM technique and the K-Means clustering algorithm. In addition to the several studies above, there are YOLO algorithms that several researchers use to detect fashion and apparel products.

2.2 Growth of E-commerce

E-commerce generally refers to the purchasing and selling of goods and services electronically using internet-based media. Additionally, E-commerce can be defined as a business activity that utilizes electronic technology to connect businesses, consumers, and the general public through electronic transactions and the electronic exchange or sale of goods, services, and information(Boysen, de Koster, & Weidinger, 2019). Shopee is one of the most widely used e-commerce platforms in Southeast Asia. Therefore, in this study, an analysis of this e-commerce will be carried out.

The exciting opportunity presented by Internet technology has stimulated the growth of online e-commerce platforms that specialize in fashion, music, food, electronics, and travel. Nowadays, E-commerce is widely regarded as the most convenient method of purchasing, as it enables us to limit virus spread and maintain a safe and comfortable environment. The increase and spread of COVID-19 have resulted in a global pandemic, estimated by the World Health Organization, with 5 million cases since May 2020 (Ranjan, Misra, & Yadav, n.d.). Online e-commerce platforms have stimulated strong competition among e-commerce businesses as they compete for new customers (Chaparro-Peláez, Hernández-García, & Urueña-López, 2015). Additionally, researchers conduct research to ensure that customers' trust is fully realized. There are several significant issues associated with e-commerce, the majority of which are faced by customers. These issues include security and quality, but we have not yet achieved an ideal world of secure transactions, owing to everyone's access to the internet (Sharma, 2020).

E-commerce is expected to become a new growth engine for developing countries if adoption barriers are removed (Lawrence & Tar, 2010). It is critical to ensure that such barriers are kept to a minimum for small and medium-sized businesses (SMEs) (Amornkitvikai & Lee, 2020). E-commerce businesses are classified into four categories: Business-to-Business (B2B), Business-to-Consumer (B2C), Consumer-to-Consumer (C2C), and Consumer-to-Business (C2B), as shown in Figure 2.1, based on their target customer, available resources, and seller and customer capabilities (Malhotra, 2021). And at the moment, academics are working to increase the quality of all sorts of e-commerce. In this thesis, we present data from the Shopee (Thailand) website and examine it qualitatively. The acquired qualitative data were primarily used for content analysis and thematic analysis.

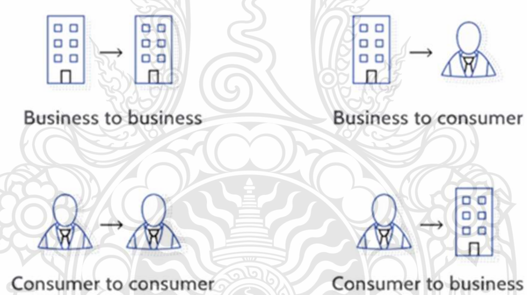


Figure 2. 1 Four Types of E-commerce

2.3 Data Collection Methods

The term "data" refers to a collection of facts, statistics, objects, symbols, and events culled from a variety of sources. Organizations gather data in order to make more informed judgments. Without data, organizations would struggle to make sound decisions, which is why data is collected at various points in time from a variety of audiences. While data is a significant asset to any company, it is useless unless analyzed or processed to get the intended outcomes. There are two sorts of data collecting methods: primary data collection methods and secondary data collection methods.

Primary data is gathered through first-hand experience and has not been used previously. The data acquired through preliminary data-gathering methods are accurate and specific to the research's purpose. Secondary data is data that has previously been used. The researcher might collect data from both internal and external sources within the organization. Secondary data gathering approaches, like primary data collection, can include both quantitative and qualitative methodologies. Secondary data is more readily available and thus less time-consuming and costly to obtain than preliminary data. However, the authenticity of the data acquired via secondary data collection methods cannot be validated.

The primary data-gathering approach will be used in this study. Methods for collecting preliminary data can be classified into three categories: quantitative, qualitative, and mixed approaches of two ways. When it comes to data collection and analysis, quantitative research focuses on numbers and figures, whereas qualitative research focuses on words and meanings. Both are critical for acquiring various types of knowledge.

The use of numbers and graphs characterizes quantitative research. It is used to verify or disprove theories and hypotheses. This research form can be used to establish a topic's generalizable facts. When a researcher is expected to quantify a problem or address the "what" or "how many" parts of a research subject, quantitative data are used. It is data that can either be counted or compared on a numeric scale. Experiments, observations recorded as numbers, and surveys with closed-ended questions are all examples of common quantitative procedures.

Qualitative research is verbal in nature. It is utilized to comprehend concepts, ideas, and experiences. This form of study enables us to gain in-depth insights into poorly understood subjects. It is gathered through questionnaires, interviews, and observation and frequently takes the form of narratives.

Mixed methods research uses quantitative and qualitative methodologies to address a research subject. Due to the fact that mixed methods studies incorporate the advantages of both quantitative and qualitative research, they can provide a more complete picture than a standalone quantitative or qualitative study. Mixed methods research is frequently employed in the behavioral, health, and social sciences, particularly in multidisciplinary settings and detailed situational or societal studies.

2.3.1 Qualitative Analysis Approaches

Qualitative research is frequently used to assist a researcher in developing a rich and nuanced understanding of a particular phenomenon (Lester, Cho, & Lochmiller, 2020). Several methods are available to analyze qualitative data. The most commonly used data analysis methods are shown in Figure 2.2.

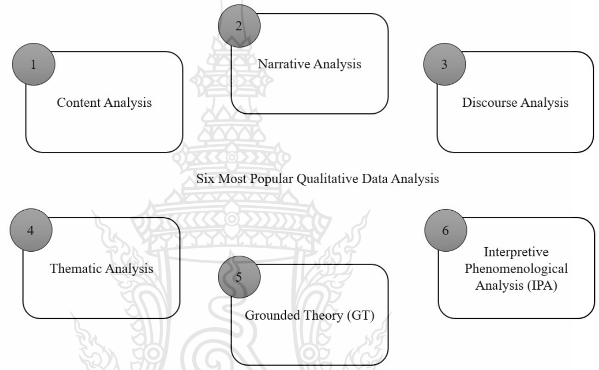


Figure 2. 2 Six Most Common Qualitative Data Analysis

2.3.2 Qualitative Content Analysis

The origins of content analysis date all the way back to the 1950s study of mass communication; Berelson published the first textbook on the subject in 1952 (*International JOurnal of Sales, Retailing & Marketing*, n.d.). Content analysis is a type of qualitative data analysis technique used to determine the presence of specific words, themes, or concepts in qualitative data (a set of text, spoken, or visual). Content analysis enables researchers to quantify and analyze the presence, meaning, and relationship of particular words, themes, or concepts (Erlingsson & Brysiewicz, 2017). The purpose of this study is to compare and validate the accuracy of the seller's image and category selections on the Shopee application.

2.3.3 Qualitative Thematic Analysis

Thematic analysis is a type of qualitative data analysis that entails poring over large data sets (such as transcripts of in-depth interviews or focus groups) and identifying meaningful patterns. Thematic analysis is extensively used in psychology, tourism, social media, and marketing (Erlingsson & Brysiewicz, 2017). This study employed thematic analysis to validate product titles that included category names. Thematic analysis was used in this study to verify product titles that contained category names. This analysis was conducted on the category 'others.'

2.4 Machine Learning

Due to recent technological advancements in the digital world, the new and most modern E-Commerce sites are leveraging the next generation of Artificial Intelligence (AI) with its sub-components Machine Learning (ML) and Deep Learning (DL) and their technical integrations into most modern E-Commerce sites. Machine Learning is a subset of AI. Machine learning is a term that refers to the process of learning from prior experience in order to improve future performance. Figure 2.3 represents the relationship between deep learning, machine learning and artificial intelligence at work. Tom Mitchell offers a modern definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves as a result of experience E." (Mitchell, 1997).

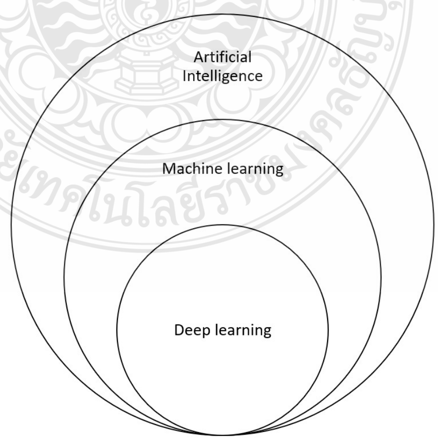


Figure 2. 3 Artificial Intelligence, Machine Learning, and Deep Learning at work

In general, there are three types of machine learning.

- Supervised learning: The machine learning algorithm is trained on labeled data. While accurate data labeling is required for this method to work, supervised learning is extremely powerful when used appropriately. In supervised learning, the labels enable the algorithm to determine the precise nature of any relationship between two data points.
- Unsupervised learning: The advantage of unsupervised machine learning is that it can be used with unlabelled data. This eliminates the need for human labor to make the dataset machine-readable, allowing the program to work with much larger datasets.
- Semi-supervised learning: Semi-supervised learning is a subset of machine learning in which models are trained using a small amount of labeled data and a large amount of unlabelled data. This approach to machine learning combines supervised and unsupervised learning. Supervised learning makes use of labeled training data; unsupervised learning makes use of unlabelled training data.
- Reinforcement learning: Reinforcement learning is directly inspired by how humans learn from data in their daily lives. It incorporates an algorithm that self-improves and learns from new situations via a trial-and-error approach.

2.5 Deep Learning

Deep learning is a subclass of machine learning that comprises neural networks with three or more layers. These neural networks seek to mimic the activity of the human brain, allowing them to "learn" from vast quantities of data. Despite the fact that a neural network with a single hidden layer can still produce approximations, multiple hidden layers can help to improve and enhance the network for precision. Deep learning drives many artificial intelligence (AI) applications and services that strengthen automation by executing analytical and physical activities without human intervention. Models of machine learning and deep learning models are also capable of several types of learning, typically characterized as supervised learning, unsupervised learning, and reinforcement learning.

Multiple layers of interconnected nodes modify and optimize the prediction or categorization in deep neural networks. This succession of computations is referred to as forward propagation. Visible layers are the input and output layers of a deep neural network. The input layer is where the deep learning model receives data for processing, whereas the output layer is where the final prediction or classification is formed. Backpropagation alters the weights and biases of the function by traveling backward through the layers in an effort to train the model using methods such as gradient descent to quantify errors in predictions. Forward propagation and backpropagation enable a neural network to generate predictions and adjust for errors. The algorithm's precision continually improves over time.

However, deep learning algorithms are highly sophisticated, and there are a variety of neural network designs to solve particular challenges or datasets. Utilized primarily in computer vision and image classification applications, convolutional neural networks (CNNs) can recognize characteristics and patterns within an image, enabling tasks such as object detection or recognition. In 2015, CNN defeated a human in an object identification competition for the first time. Recurrent neural networks (RNNs) are often utilized in natural language and speech recognition applications using sequential or time-series data.

Object detection is a deep learning method where humans, things, and cars can be detected as objects in images and videos. It detects the object with bounding boxes in the image. The difference between object detection algorithms and classification algorithms is that in detection algorithms, we try to draw a bounding box around the object of interest to locate it within the image. There could be many bounding boxes representing different objects of interest within an image. The main reason why a standard convolutional network cannot solve this problem is that the length of the output layer is variable.

Deep learning object detectors are classified into two types: detectors with one and two stages. Then, two stages are required to complete the two-stage object detection process. Two-stage detectors can be separated into two phases by the RoI (Region of Interest) pooling layer (Jiao et al., 2019). The R-CNN series are representative networks based on candidate regions (Fu et al., 2020). R-CNN(Girshick, Donahue, Darrell, & Malik, 2014), Fast R-CNN(Girshick, 2015), Faster R-CNN(Ren, He, Girshick, & Sun, 2017) and Mask R-CNN(He, Gkioxari, Dollar, & Girshick, 2017) are all two-stage detectors. For instance, the best-known, Faster R-CNN(Ren et al., 2017) begins with a step called RPN, which stands for Region Proposal Network, which is used to locate the potential bounding boxes. In the second step, features are extracted from each bounding box using an RoI pooling procedure for the classification and bounding box regression tasks. As a preliminary phase, proposals and regional classification are made (Lohia -Guise, n.d.) Fast/Faster RCNN (two-stage detector) is preferable for immediate information access since the bounding box, and object class estimation processes are carried out in tandem (Fujii & Kawamoto, 2021).

The purpose of this study is to detect and categorize fashion trends. This article makes three significant contributions: In contrast to previous approaches, deep learning enables us to extract features from low to high-level images. The deep learning image features are more representative than the handmade features. As a result, the researchers eventually shifted their focus to DCNNs. Object detection aims to identify distinct item categories and precisely locate category-specific objects using bounding boxes. During the last two decades, it has been widely accepted that developments around object detection occurred in two distinct historical periods. There was traditional object detection prior to 2014 and deep learning-based object detection following 2014 (Y. Xiao et al., 2020).

Due to the processing cost of region proposal approaches and their incompatibility with smartphones and other wearable sensors, numerous researchers have pioneered the development of single-stage detection pathways (Arulprakash & Aruldoss, 2021). This is because single-stage detectors are faster in detecting speed and produce more reliable results in terms of accuracy in real-world scenarios. As a result, a single-stage detector is optimal for vast volumes of data, such as those seen in clothing fields. OverFeat (Sermanet et al., 2013), SSD(Leibe, Matas, Sebe, & Welling, 2016), RetinaNet (Lin, Goyal, Girshick, He, & Dollár, 2017), Cornernet (Law & Deng, 2018) and YOLO (Redmon, Divvala, Girshick, & Farhadi, 2015a), which are all one-stage detectors, have garnered attention for their speed of processing. Certain parameters, including localization, inference speed, and accuracy, must be carefully examined while designing and selecting one of these detectors.

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) is one of the most advanced areas in machine learning and image processing. The name is taken from the analogy of human nerves and how the human brain works from convolutional neural networks [15]. Its history dates back to the 1990s when LeCun et al. used a gradient-based machine learning algorithm on CNN, and they got successful outputs of classifying digits written by hand [16]. In deep learning, CNN is a class of artificial neural networks consisting of an input layer, many hidden layers, an output layer, and millions of parameters that can learn complex objects and patterns. The hidden layers include layers that perform convolutions. A detail of what CNN does is that we take the image, pass it through a series of convolutional layers with filters, pooling, and fully connected layers and get output by applying the SoftMax function to classify an object with values between 0 and 1. The result can be a single class or a probability of types.

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. The role of the Convolution Network is to reduce the images into a form that is easier to process without losing features critical for getting a good prediction. It is a mathematical operation that takes two inputs, such as an image matrix and a filter which is the element involved in carrying out the convolution operation in the first part of a convolutional layer. Convolution of an image with different filters can perform various functions such as edge detection, blur, and sharpening.

The filter, kernel, or feature detector is the 3 x 3 matrix. The matrix formed by sliding the filter over the image and computing the dot product is called Convolved

Feature, Activation Map, or Feature Map. The number of filters used in the convolution operation is called depth.

2.7 You only look once (YOLO)

YOLO came onto the computer vision scene with the seminal 2015 paper by (Redmon, Divvala, Girshick, & Farhadi, 2015b), "You Only Look Once: Unified, Real-Time Object Detection," and it got great attention from computer vision researchers. You only look once (YOLO) is a state-of-the-art, real-time object detection system. A Pascal Titan X processes images at 30 FPS and has an mAP of 57.9% on the COCO test-dev [18]. It is so fast that it has become almost a standard way of detecting objects in the field of computer vision. Previously people were using sliding window object detection. YOLO was invented, which outperformed all the previous object detection algorithms. Born of a new object detection approach after multiple improvements of object detection that uses bounding boxes of object detection instead of classifying whole image pixels (that have been used in object detection algorithms before), also is a real-time object detection because of the improvement in speed of recognition, the previous approaches did not provide it, they were slower than YOLO, so it cannot detect real-time videos while YOLO did that, in YOLO algorithm takes a single look to the image and extract the objects that made YOLO much faster but this effects the accuracy too (Khalaf & Akbulut, 2021).

Object detection is one of the challenging problems in computer vision in order to detect what objects are inside an image and where they are. YOLO can predict more than one object in the images by showing bounding boxes for each object. YOLO algorithm uses a single neural network to the image, then divides the images into regions and predicts bounding boxes and probabilities for each region. YOLO trains on full images and directly optimizes detection performances. YOLO algorithms outperform other detection methods when trained on natural images and tested on artwork. Figure 2.4 represents the convolutional neural network architecture of the YOLO model.

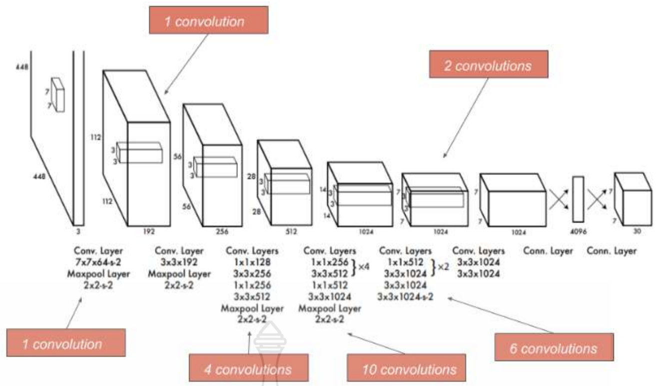


Figure 2. 4 CNN architecture for YOLO

First, the image is divided into grid cells. Each grid cell forecasts and provides confidence scores for B bounding boxes. Cells predict class probabilities in order to determine the class of each object. All predictions are made concurrently with the aid of a single convolutional neural network. The final detection will produce unique bounding boxes that perfectly fit the objects. YOLO algorithm works using the following three techniques:

- Residual blocks
- Bounding box regression
- Intersection Over Union (IOU)

2.7.1 Residual Blocks

Residual Blocks are skip-connection blocks that learn residual functions in reference to the layer's inputs rather than unreferenced functions. They were integrated into the ResNet design as a component (He, Zhang, Ren, & Sun, 2015a). Residual Network (ResNet) is a specific type of neural network introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. The performance degrades by adding more layers on top of the network in traditional neural networks. This problem of training very deep networks caused the introduction of ResNet or residual networks, which are made up of Residual Blocks. There is a direct connection that skips some layers in between, called a skip connection, which is the core of residual blocks. Figure 2.5 shows the building block of the residual learning approach.

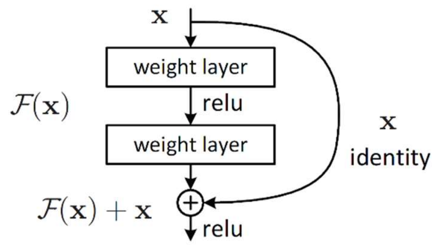


Figure 2. 5 Residual learning: a building block

The skip connections in ResNet solve the vanishing gradient problem in deep neural networks. This is because when the network is too deep, the gradients from where the loss function is calculated easily shrink to zero after several chain rule applications. With ResNets, the gradients can flow directly through the skip connections backward from later layers to initial filters.

2.7.2 Bounding box regression

A bounding box is a rectangle that serves as a location for objection and creates a collision box for that object. Each bounding box in the image consists of width(bw), height(bh), class of the object(c), and bounding box center (bx, by). Regression is a statistical technique used in finance, investing, and other fields to determine the strength and nature of the relationship between a single dependent variable and a series of independent variables. YOLO uses the bounding box regression to predict objects' height, width, center, and class as shown in Figure 2.6.

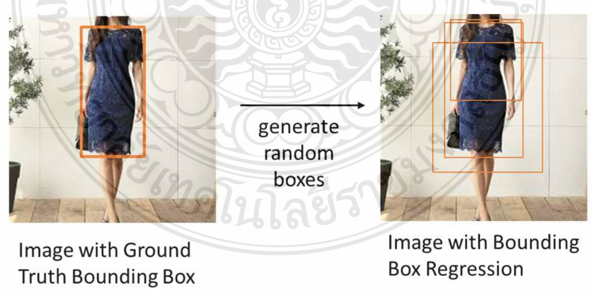


Figure 2. 6 Bounding Box Regression

2.7.3 Intersection over union (IOU)

Intersection over union ensures that the predicted bounding boxes of the objects are identical to their actual bounding boxes as shown in Figure 2.7. It is a number from 0 to 1 that specifies the amount of overlap between the predicted and ground truth bounding box. This phenomenon eliminates superfluous bounding boxes that do not correspond to the objects' properties. The IOU equals one if the predicted bounding box is the same as the real box. The IOU is equal to zero if there is no overlap between the boxes.

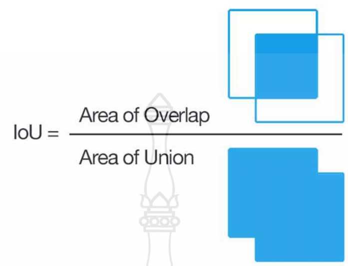


Figure 2. 7 Intersection over union (IOU)

2.7.4 Development of YOLO versions

Four versions of YOLO have been released, ranging from YOLO v1 to YOLO v4. In 2015, YOLO's first version (YOLOv1) (Redmon et al., 2015b) used a single neural network to predict numerous classes' bounding boxes and related class probabilities. YOLOv1 converts object recognition into regression difficulties. YOLOv1's primary objective is to build a CNN network capable of predicting tensors. On Pascal VOC2007, YOLOv1 analyzes images at 45 frames per second (fps) rate, which is twice to nine times quicker than Faster R-CNN. Version 2 of YOLO (YOLOv2, or YOLO9000) (Redmon & Farhadi, 2016) is a simplified version of YOLO's single-stage real-time object detection model. It preserves YOLOv1's speed advantage while aiming to boost the mAP value. It improves on YOLOv1 in several ways, including the use of Darknet-19 as batch normalization, backbone, a high-resolution classifier, and anchor boxes to predict bounding boxes. It is still incapable of generalizing to novel or unexpected aspect ratios or configurations of objects. Additionally, YOLO's loss function treats errors uniformly in small and large bounding boxes and is not flawless. Then, a new version of YOLOv2, dubbed YOLOv3 (Redmon & Farhadi, 2016), was developed that uses logistic regression to anticipate the objectiveness score of each bounding box and modifies how cost functions are derived. YOLOv3 calculates each label's binary cross-entropy loss. This may result in a reduction in computing complexity by omitting SoftMax features. (Sichuan Institute of Electronics & Institute of Electrical and Electronics Engineers, n.d.) proved that

YOLOv2 outperforms Faster R-CNN in the identification of fashion clothes. By combining the previous research, it was determined that the YOLO method is more suited for detecting fashion items than the two-stage detection techniques. YOLOv3 is three times quicker than the previous version but has a more significant positioning error. However, it significantly improves tiny object detection. In 2020, YOLOv4 (Bochkovskiy, Wang, & Liao, 2020) was released, built on the Darknet framework, a freely available neural network framework written in C and CUDA. YOLOv4 takes advantage of the PANet (Path Aggregation Network). This distinction enables the YOLOv4 to be quicker and more precise than the YOLOv3.

Researchers are continually trying to enhance the YOLO standard algorithm for one-stage processes that are as efficient at identifying the position of objects as two-stage approaches. In 2021, (Chai, Ta, Ma, & Zhi, 2021) suggested the "YOLO with a bigger effective receptive field (ERF-YOLO)" algorithm, which would lower the number of model parameters while keeping the same performance. Then (Lee & Lin, 2021) presented YOLOv4-TPD (YOLOv4 Two-Phase Detection), which included a two-phase approach for detecting fashion clothes and an image enhancement method based on CLAHE. This algorithm has more parameters when compared to ERF-YOLO. Their primary focus areas include jackets, tops, slacks, skirts, and handbags. The two-phase training technique resulted in mAP outperforming the original YOLOv4 model in two-phase transfer learning. Transfer learning is adapting previously acquired information and abilities to new activities.

2.8 TensorFlow Lite

A number of popular deep learning frameworks, including TensorFlow Mobile (TFM), TensorFlow light (TFL), OpenCV, and Qualcomm Snapdragon, have been adapted to the Android operating system in response to its rising popularity. These platforms will execute the inference appropriate for mobile phone use in situations where there is low or non-existent connectivity. Each of these prominent frameworks has its own advantages and disadvantages, which should be considered when selecting the optimal architecture for object identification. Google has built the machine learning platform TensorFlow Mobile. TensorFlow Lite is the lightweight version of TFM for restricted neural network surroundings with decreased latency and increased productivity. TFL requires a converter to transform the neural network configuration file and its associated weights into a flat buffer model (.tflite).

In contrast, TFL is more adaptable than TFM, being able to execute machine learning algorithms on all on-board processors. Similar to TFM and OpenCV, TensorFlow Lite is able to straightforwardly execute the neural network on a CPU method, whereas TFL provides faster time for inference. In addition, TFL can outsource a portion or all of an algorithm's execution to the mobile GPU. GPU delegation and the android neural network API (NNAPI) are the two major approaches used for delegation. This platform is compatible with its own drivers, allowing direct manipulation of onboard CPUs (CPU, GPU, and DSP). This will lead to enhanced productivity.

CHAPTER 3

RESEARCH METHODOLOGY

This chapter provides an overview of the research methods used in the study. It describes the data collection methods and tools to conduct this research and provides an explanation for the method for data analysis. This study also includes a description of our modified object detection for our collected dataset and provides an explanation for its application.

3.1 Data collection methods and tools

3.1.1 Shopee Data Scraper

This investigation began with data collection. Shopee Data Scraper (Data Sunday) is a browser plugin that collects data from the Shopee platform, such as product titles, product categories, product lists, and product photos. Figure 3.1 represents how to collect the Shopee product list from Data Sunday. This study included seven categories, each with eight hundred and six product listings and five photos. Due to the fact that Shopee.co.th displays product titles in Thai, this research used the Google Translate tool on a Google Sheet to extract solely English-language product titles. We collected 100 for the shirt, 100 for the dress, 102 for the jumpsuit, 101 for the skirt, 101 for the denim, 101 for trousers, and 201 for other categories accordingly.

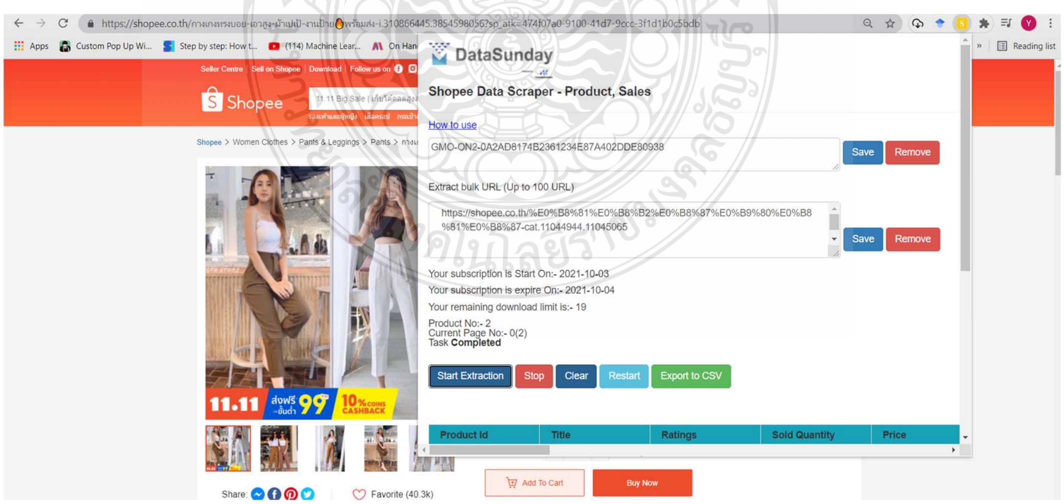


Figure 3. 1 “DataSunday” tool to collect Shopee Product List

3.1.2 Google Colab

Google Colab (short for Collaboratory) is a free platform provided by Google that enables users to write Python code. Colab is essentially Google's Jupyter Notebook replacement. Colab's advantages over Jupyter include zero-configuration, unrestricted access to GPUs and CPUs, and seamless code sharing. Colab is being used by an increasing number of people to take advantage of high-end computing resources without being constrained by their cost. The first step in any data science project is data loading. Frequently, data loading into Colab requires additional setups or coding.

3.1.3 LabelImg

Labeled data is critical for successful machine learning, and this is true for computer vision as well. LabelImg(Tzutalin, 2015) is a free, open-source tool for labeling images graphically. Figure 3.2 shows the user interface of the LabelImg tool. It is written in Python and features a graphical user interface powered by QT. It's a simple, cost-effective method of labeling a few hundred images to test your next object detection project. LabelImg supports VOC XML or YOLO text file labelling.

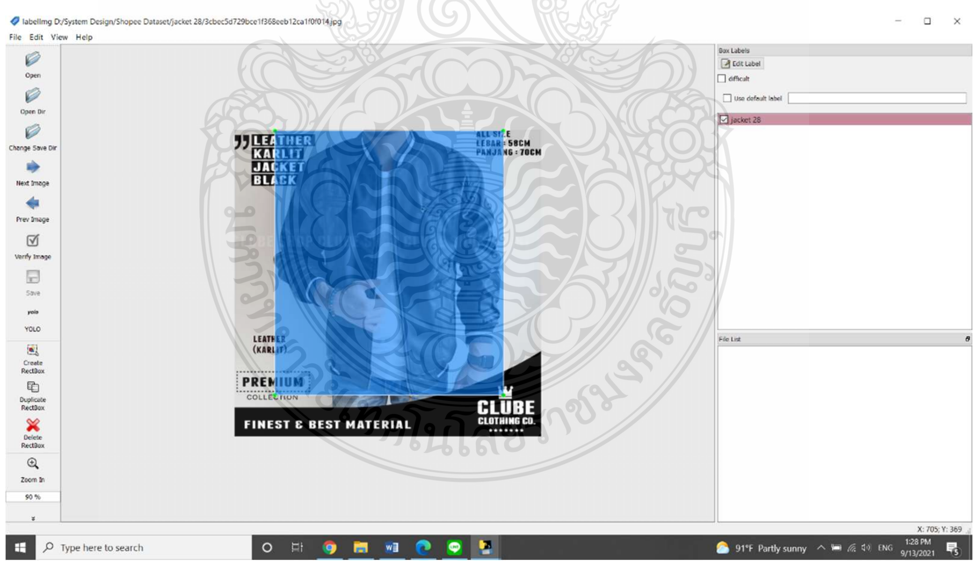


Figure 3. 2 LabelImg User Interface

3.1.4 Darknet Framework

Darknet is a free and open-source framework for neural networks written in C and CUDA. It is lightweight and simple to install, and it supports both CPU and GPU computation. The source code is available on GitHub. Darknet requires only two optional dependencies: OpenCV for a broader range of supported image types and CUDA for GPU computation. Neither is required, but users may begin by installing the base system, which has been tested exclusively on Linux and Mac computers. You Only Look Once (YOLO) is a state-of-the-art, real-time object detection system included in the framework. It processes images at 40-90 frames per second on a Titan X and has an mAP of 78.6 percent on VOC 2007 and 44.0 percent on COCO test-dev. Users can use Darknet to classify images submitted to the ImageNet 1000 class challenge. Darknet displays information as it loads the configuration file and weights the image then classifies it and prints the image's top ten classes. Darknet is capable of handling recurrent neural networks, which are powerful models for representing data that changes over time, without requiring CUDA or OpenCV. Additionally, the framework enables users to experiment with game-playing neural networks.

3.2 Data Analysis methods and tools

This study employs a qualitative methodology. Quantitative research methods are those that make an attempt to correctly quantify behavior, knowledge, attitudes, or opinions (Bauer & Scheim, 2019a). Quantitative research methods are frequently employed in a variety of investigations due to their usefulness for testing models or hypotheses (Queirós, Faria, & Almeida, n.d.-a). This research employs two methodologies: content analysis and thematic analysis. The second stage is content analysis, which involves comparing photos and product categories. The content analysis method is used to study the categories "shirts," "dresses," "jumpsuits," "skirts," "denim," and "trousers." When we conduct the content analysis on these categories, we examine both the category and the product photos to determine whether or not the category was accurately identified. Manual verification is performed with the use of human visuals. True if the image matches the category; false if it does not.

The thematic analysis was done, with an emphasis on the category 'others.' This is because several terms in this category are synonymous with terms in other categories. Occasionally, merchants are uninformed of the category in which their product belongs and choose another. We categorize products based on their title and image. As has been noted, product titles frequently incorporate category words or synonyms for category words. As a result, we create product categories based on the title's category word and analyze which categories fall within the category "other." Once all stages have been completed, the final step is to analyze the results. The study's conclusion will be based on these findings. The conceptual framework of data analysis methods is shown in Figure 3.3. Two methodologies are used in this study: content analysis and thematic analysis.

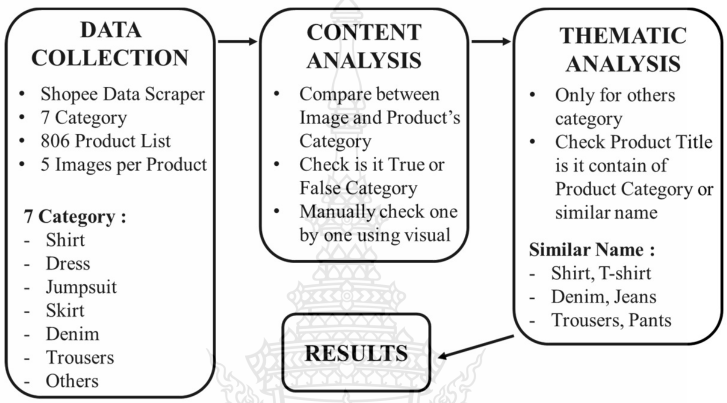


Figure 3. 3 Conceptual Framework

This study takes a qualitative approach. Quantitative research methods are those that attempt to quantify behavior, knowledge, opinions, or attitudes accurately (Bauer & Scheim, 2019b). Due to their suitability for testing models or hypotheses, quantitative research methods are widely used in a variety of studies (Queirós, Faria, & Almeida, n.d.-b). This research employs two methodologies: content analysis and thematic analysis. Fig. 1 depicts the conceptual framework for this research.

The initial stage of this research was data collection. The Shopee Data Scraper (Data Sunday) browser extension is used to collect data from the Shopee platform, including product titles, product categories, product lists, and product images. Seven categories were used in this study, with eight hundred and six product lists and five images for each. Due to the fact that Shopee.co.th displays product titles in the Thai language, this research uses the Google Translate function on a Google Sheet to extract only titles in the English language.

The second stage is content analysis, during which images and product categories are compared. The categories of "shirts," "dresses," "jumpsuits," "skirts," "denim," and "trousers" are all analyzed using the content analysis method. When we utilize content analysis in these categories, we look at the category as well as the images of the product to assess whether or not a category has been correctly classified. Manual verification is carried out using human visuals. If the image matches the category, it is true; if it does not, it is false.

Thirdly, a thematic analysis was conducted, focusing exclusively on the 'others' category. This is because some of the terms in this category overlap with those in other categories. Occasionally, sellers are unaware of their product's category and opt for another. We use the product's title and image to identify this category. In many cases, product titles incorporate category words or synonyms of category words, as has been noticed. As a result, we construct product categories based on the category word that appears in the title and investigate which categories are included in the category "other". After completing all stages, the final step is to analyze the results. These findings will serve as the study's conclusion.

3.3 A Semi-Supervised Learning Approach for Automatic Detection and Fashion Product Category Prediction using FC-YOLOv4

This section discusses dataset collection and pre-processing for our detection, a description of traditional YOLOv4 architecture, a modified part of our proposed FC-YOLOv4, and the procedure description of our semi-supervised learning for clothing identification and classification.

3.3.1 Dataset Collection and Pre-Processing

For category detection of fashion clothing datasets from E-commerce, some problems and issues can be considered as follows.

- As shown in Figure 3.4(a), depending on the point of view, the same clothes can be considered in different categories (e.g., skirt or dress category), and different clothing can be considered in the same category.

- Figure 3.4(b) illustrates that the clothing form can be simply modified by stretching, folding, hanging, and changing the model position.

- As demonstrated in Figure 3.4(c), the same category can look different due to varying perspectives and illumination, cluttered backdrops, and being partially obscured by other things or people.

(a) (b) (c)

Figure 3. 4 Visualization of the fashion categories

Based on these problems, we require the product dataset directly collected from multiple sellers on E-commerce platforms to aid in fashion category detection. From 2012 to 2020, there are 51 datasets in the fashion business, according to the study (Mameli et al., 2022). Datasets for clothes Parsing, clothing landmark identification, product retrieval, clothing generation, clothing recommendation, and fashion classification are covered. Only two datasets were developed for fashion categorization, according to the research(Mameli et al., 2022): Fashion MNIST (H. Xiao, Rasul, & Vollgraf, 2017), which originates from the shopping website named Zalando, and CBL (K. H. Liu, Liu, & Wang, 2021), which was created from 25 clothing brands, and the other datasets used for object detection task are (de Souza Inácio & Lopes, 2020; Jia et al., 2020; Tiwari, Bhatnagar, Tung, & Pons-Moll, 2020; Wu et al., 2019; Yamaguchi, Hadi, Luis, Ortiz, & Berg, n.d.; Yang et al., 2020). There are not many fashion classification datasets developed, particularly those gathered from E-commerce shopping platforms.

In this study, we proposed data acquired from Shopee E-commerce Thailand between October and December 2021. The images of this study were collected using Shopee Data Scraper, a tool for extracting product data from the Shopee website. A total of 663 photos of products were obtained from the website as dataset A after deleting unnecessary images. We gathered images in various resolutions, with and without background noise, and images captured by vendors. On the other hand, we gathered E-commerce photos from Google images according to our categories, and a total of 5,772 images were collected as dataset B. The images in Datasets A and B range in aspect ratio from 225*225 pixels to 800*800 pixels, and we aggregated the two datasets as Shopee Image Dataset (Thailand) (Yamin Thwe, 2022). Our dataset is published to the IEEE Data Port with the required annotation bounding box result files and picture data for usage by other researchers. Figure 3.5 shows the sample images of the Shopee Image Dataset. Before training the model, it is essential to annotate the training datasets with the YOLO format manually.

Figure 3. 5 Samples of our Dataset

The process of labeling photos in a dataset in terms of training a machine learning model is called image annotation. Labeling pictures is critical since it informs the training model about the objects that need to be detected. Our study accomplished this task using the LabelImg(Tzutalin, 2015) tool. LabelImg is a free and open-source program for annotating graphical images. It creates bounding box annotations in the PASCAL VOC format. Additionally, it supports the YOLO and CreateML formats. Each XML file specifies the item's class, coordinates, height, and breadth. Equation (1) illustrates how to annotate our image. L is a collection of the image's bounding boxes, Bn is the bounding box for the n^{th} object. When two items of clothing overlap in an image, we only draw the overlapping object if it comprises less than 50 percent of the other image, as shown in Figure 3.6. Afterward, it allows the model to differentiate those things into new or never-seen-before objects.

$$L = \{ L_{B1}, L_{B2}, \dots \dots \dots, L_{Bn}\}, B_n \cap B_{n+1} < 0.5 \qquad (1)$$

Figure 3. 6 Image Labelling Process

In this research, when we choose a category, we select two categories that best match each pattern. Pants, dresses, skirts, hoodies, and jackets are included. To identify the categories of accessories, we chose categories that are small and contain similar patterns, such as Bracelets. When gathering photographs, we collected various types of images uploaded by vendors in order from the website's list. Typically, when sellers submit a product, they include a single design as well as variations of the product in the photograph. We annotated photos with a single bounding box and images with multiple bounding boxes. Table 3.1 displays the number of photos and bounding boxes within each category for the total of Dataset A and Dataset B. The collected image collection includes a maximum of 18 bounding boxes per image, as described in Table 3.2.

Table 3. 1 Number of pictures and bounding boxes per category

Categories	Number of images	Number of bounding boxes
Pants	741	1,212
Dress	699	1,048
Hoodie	1,059	1,843
Jacket	684	977
Skirt	933	1,685
Necklace	219	238
Belt	343	514
Ring	483	990
Earrings	159	167
Bracelet	452	554
Total	5,772	9,228

Table 3. 2 Number of images according to the bounding boxes per category

Categories	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Pants	428	173	76	19	12	4	3	4	-	2	-	-	-	-	-	-	-	-

Dress	485	142	41	11	5	7	4	-	-	-	-	-	1	-	-	-	-	-
Hoodie	725	177	54	46	25	18	5	4	6	2	-	-	2	-	-	-	-	-
Jacket	561	62	39	22	9	1	1	-	1	-	-	-	-	-	-	-	-	-
Skirt	605	177	63	18	19	13	16	7	4	3	1	2	1	1	-	-	-	-
Necklace	214	9	-	-	-	6	-	-	9	-	-	-	-	-	-	-	-	-
Belt	280	40	45	44	50	12	14	-	18	-	11	-	-	-	-	-	-	-
Ring	328	103	96	111	65	60	62	48	18	28	-	12	13	28	-	-	-	18
Earrings	155	4	-	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Bracelet	413	28	3	25	15	6	8	8	36	12	-	-	-	-	-	-	-	-

Each category contains a variety of designs, colors, and patterns. Figure 3.7 represents the visualization of the distribution of five fashion clothing from our obtained dataset. Our collection includes product photographs with indoor, outdoor, solid, and other backgrounds. We collected photographs from many perspectives, including back, front, side, and others. Some models are photographed while standing, while others are photographed while sitting and promoting the product. There are also products with hanging styles, especially in the Hoodie category. Due to the fact that it is a garment, we have collected product images with various colors, lengths, and patterns. Because the majority of products are advertised with models, the models' accessories and hand positions cover a portion of the product. Lower-body garments, such as pants and skirts, are sometimes obscured by upper-body garments, resulting in partial occlusion. Depending on the model's posture, particular pants have similar patterns to skirts. Therefore, it would be more difficult to distinguish types of clothing from others. Photographs in the ring category exclusively depict model fingers. Photos of necklaces feature the model's neck, photos of bracelets feature the model's wrist, photos of earrings feature the model's ear, and photos of belts feature jeans or a blank background.

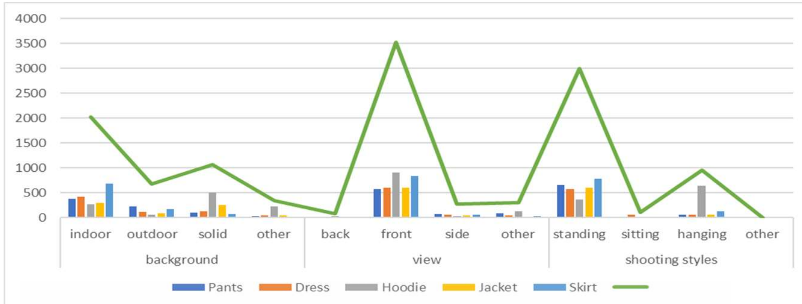


Figure 3. 7 Visualization of the Distribution for Dataset

In our research, we will also propose a Fashion category detector trained on our small dataset to identify categories with similar patterns. We utilized Google Colab Notebook Pro Version for this investigation, and our model was trained and validated on an NVIDIA-SMI 495.46 Telsa P100-PCIE server running NVIDIA Driver 460.32.03 and CUDA 11.2.

3.3.2 Traditional YOLO4 Architecture

The following section describes the architecture of the traditional YOLOv4. YOLOv4, which Alexey Bochkovskiy and the other researchers introduced, integrates the most powerful optimization approaches in the CNN domain in recent years with the traditional YOLO series. It compromises detection speed and precision, which is noticeably better than YOLOv3. Three network components comprise the traditional YOLOv4 model: backbone, neck, and head.

The "backbone" network component, which performs the role of a feature extractor, consists of a convolutional layer, a standardization layer, and an activation function. VGG16 (Simonyan & Zisserman, 2014), ResNet-50 (He, Zhang, Ren, & Sun, 2015b), EfficientNet (M. Tan & Le, 2019), CSPREsNeXt50 (C.-Y. Wang et al., 2019), and CSPDarknet53 (C.-Y. Wang et al., 2019) are currently more popular backbone detectors. We used CSPDarknet53 for our approach because it provided a greater frame rate than the others, and it greatly expands the receptive field, isolates the most important context characteristics, and results in nearly no reduction in network operating speed(Bochkovskiy et al., 2020). CSPDarknet53 includes 29 convolutional layers, 725 x 725 receptive fields, and 27.6M parameters[44].

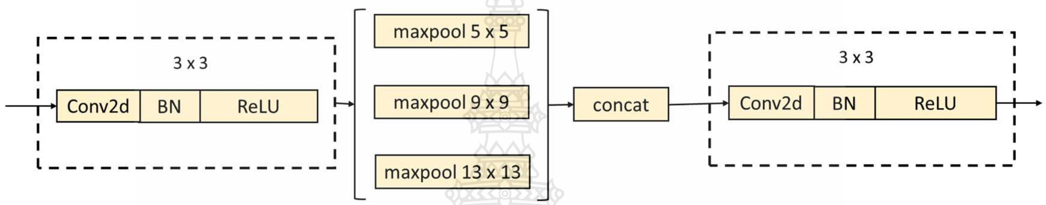
The “neck” of the object detector is used to gather feature maps between the backbone and the head. After the CSPDarknet53, we add Spatial Pyramid Pooling or SPP(He, Zhang, Ren, & Sun, 2014) blocks to enhance the receptive field. It isolates the most critical context characteristics and has a negligible impact on network operation performance. The pooling cores have a diameter of 5 x 5, 9 x 9, and 13 x 13, respectively. We use Path Aggregation Networks or PANet(S. Liu, Qi, Qin, Shi, & Jia, 2018) to aggregate parameters from various backbone levels for various detector levels. In the initial version of PANet, the current layer's information is combined with that of a previous layer to create a new vector. In the YoloV4 implementation, the new vector is formed by concatenating the input vector with the vector from the preceding layer.

The “head” of the one-stage object detector is used to conduct prediction. The bounding box and its confidence for the identified category can be determined according to the output information. It constructs final output vectors with class probabilities, objectness scores, and bounding boxes using anchor boxes. Since it is a single-stage detector without an area proposal module, it uses seven anchors. A single-stage object detector is faster but performs lower than a two-stage detector.

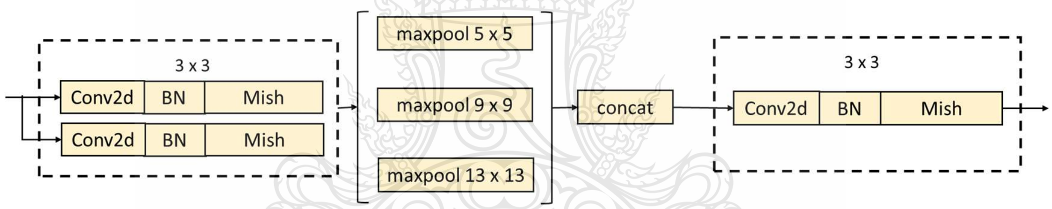
3.3.3 Modified Part of Proposed FC-Yolov4 Architecture

In traditional YOLOv4, three layers are selected from the output of the backbone network. After passing through three convolutional layers, the input is routed via an SPP network for pooling. Pooling layers enable the downsampling of feature maps by enumerating the features present in patches of the feature map. Our proposed method's design goal is to enhance the output characteristics of backbone networks. The deeper the backbone layer, the greater the risk of overfitting and the greater the error rate. As a result, we conducted research to increase the quality of the feature map depicted in the picture, as well as the model's detection, as shown in Figure 3.8 (b). The architecture of such a Neck network can achieve certain highly desirable properties.

Selecting the appropriate activation function is also critical for optimizing YOLOv4 detection accuracy. Activation functions are non-linear point-wise functions that introduce nonlinearity into the linear transformed input at the layer between the backbone and the head of a neural network. We also need to employ the optimal activation function in our FC-YOLOv4 to address computer vision issues such as object identification, segmentation, and classification. The primary objective of the activation function is to establish the non-linear relationship between the input and output variables to solve complicated issues. We used the Mish activation function instead of ReLU in our study because it significantly improves the deeper network of the model. The LeakyRelu activation function remains throughout the rest of the network. Mish and LeakyRelu activation functions can be expressed as Equations (2) and (3). The network architecture of our FC-YOLOv4 structure is shown in Figure 3.9.



(a)



(b)

Figure 3. 8 Neck network structure (a) Traditional YOLOv4 (b) our proposed FC-YOLOv4

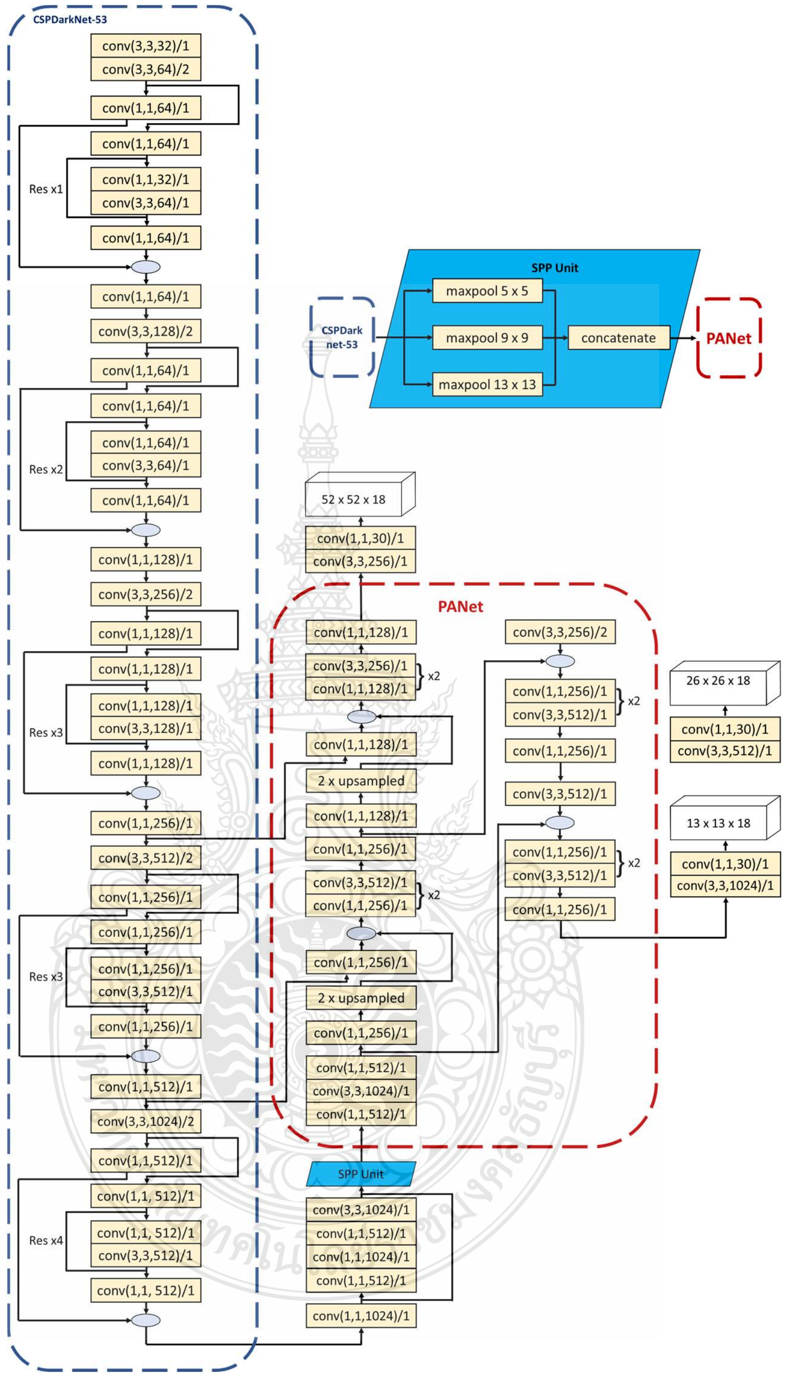


Figure 3. 9 Network architecture of FC-YOLOv4

$$y_{mish} = x\ tanh(ln(1 + e^{x})) \quad \text{……………………. (2)}$$

$$y_{Leaky\ Relu} = \begin{cases} x, & if\ x \geq 0 \\ \boldsymbol{\lambda} x, & if\ x < 0 \end{cases} \quad \text{……………………. (3)}$$

We employed max pooling in SPP Block, a pooling process that determines the maximum, or most significant, value in each patch of each feature map. We expanded the convolutional layer and executed the procedure first on the middle two feature layers. The PANet was chosen for our model because of its ability to reliably store spatial information, which aids in the proper localization of pixels for mask creation. This procedure significantly expands the effective receptive field.

We used class label smoothing to transform soft labels during training, increasing the model's robustness. In terms of the bounding box regression loss function, we use the CIoU function because it results in a faster convergence rate and superior performance than the alternatives. CIoU loss is only effective when the anticipated bounding box overlaps with the target bounding box.

3.3.4 Procedure Description

Incomplete learning trains model parameters using partial points, and prediction accuracy are significantly worse than fully supervised learning. This is because a small number of labeled points cannot sufficiently describe the general distribution of the data. As a result, employing more training data can result in more accurate predictions. As a semi-supervised learning approach, pseudo-labels may successfully utilize unlabeled data.

As shown in Figure 3.10, data collecting is the first step in this study. The traditional YOLOv4 model is developed by manually labeling and training Dataset A, which is collected from Shopee Data. Testing is carried out on dataset B, which has not been labeled, using this model. The format conversion to the YOLO format is then performed based on the findings of the label prediction. After labeling the image collection, it is necessary to feed the annotated data into the YOLOv4 model. The quantity of the training dataset is a critical aspect to consider when assessing the accuracy of the object recognition model. Even though several datasets are available, we cannot always train using them. Additionally, the data annotation process is manual, which explains why labeling a huge dataset takes so long. Pseudo-Labelling is one of the most effective methods for resolving this issue. Three types of learning exist; supervised, unsupervised, and semi-supervised. While supervised learning is the process of generating a model from labeled data, unsupervised learning is developing a model from unlabeled data. Semi-supervised learning is a strategy that entails training the model on a small set of labeled data and then predicting on a large set of unlabeled data. Pseudo-Labelling is semi-supervised learning.

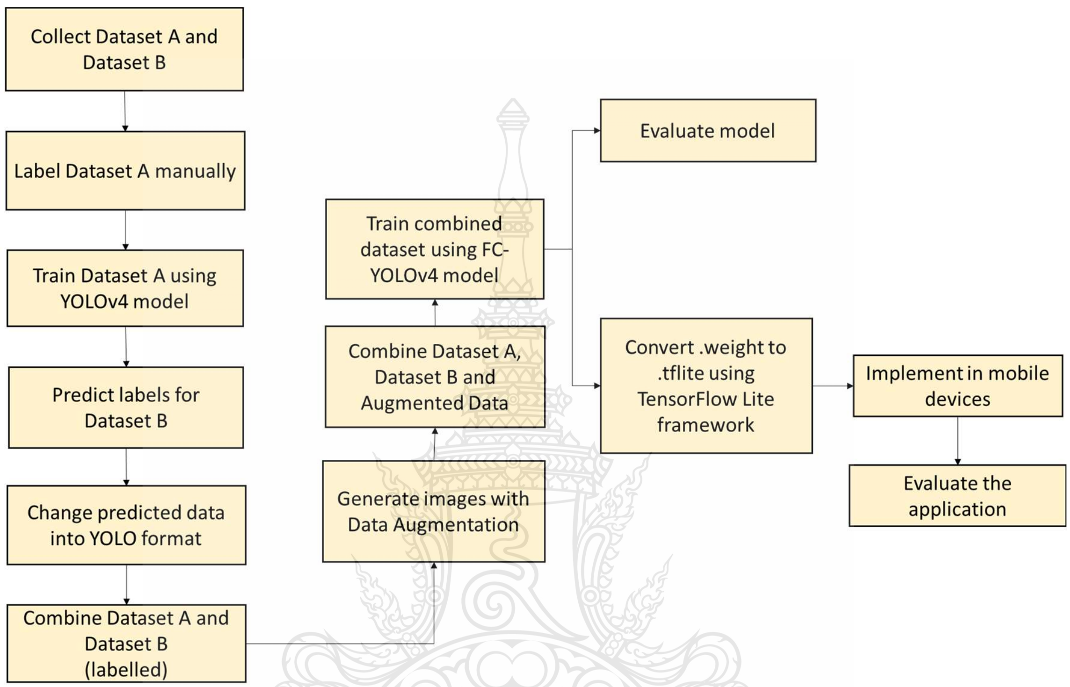
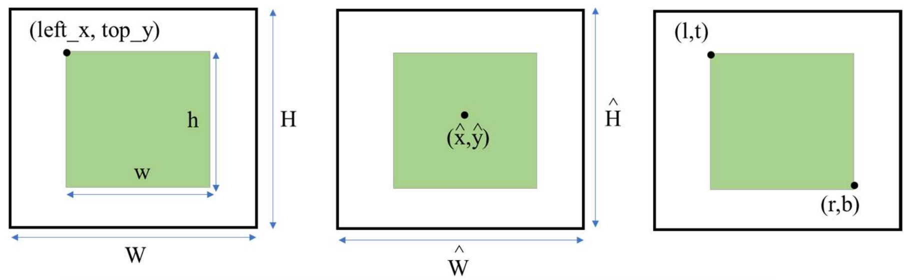


Figure 3. 10 Research Methodology

Figure 3. 11 Detection result of the YOLOv4 traditional model (Mid-length dress: 95% (left_x: 42 top_y: 13 width: 142 height: 196))



(a) (b) (c)

Figure 3. 12 Conversion from Detection Output to YOLO Form

YOLOv4 predicted the objects in the image by using the detection output format, as shown in Figure 3.11. Therefore, it must be converted to YOLOv4 format. We captured the detection result in the (.txt) format with the same filename as the image after training with dataset A using YOLOv4. Figure 3.12(a) represents the detection result of the YOLOv4 model after training. When forecasting, the YOLO model specifies the top left corner coordinates, followed by the width (w) and height (h) of the enclosing box (h). The width and height of the original image are denoted by (W) and (H), respectively. This result will be the input for Figure 3.12(b).

$$\hat{x} = \frac{left\_x + \frac{w}{2}}{W} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

$$\hat{y} = \frac{top\_y + \frac{h}{2}}{W} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

$$\widehat{W} = \frac{w}{W} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

$$\widehat{H} = \frac{h}{H} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Figure 3.12(b) is the YOLO format for retraining the model. The center point of the image is represented by $\hat{x}$ which is obtained from calculating Equations (4) and (5). While the size of the image is represented by $\widehat{W}$ and $\widehat{H}$ as the width and height of the retrain model results are obtained from Equations (6) and (7). Furthermore, the results of this retraining will be the input for drawing the bounding box in Figure 3.12(c). The estimated bounding box values must be computed in YOLO format according to the draw bounding box Equation (8), (9), (10), (11) where respectively l, r, t, and b are the left, right, top, and bottom sides of the box.

$$l = \left(x - \frac{\widehat{W}}{2}\right) * W \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

$$r = \left(x + \frac{\widehat{W}}{2}\right) * W \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

$$t = \left(y - \frac{\widehat{H}}{2}\right) * H \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

$$b = \left(y + \frac{\widehat{H}}{2}\right) * H \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (11)$$

Algorithm 1: The pseudocode of pseudo-labeling to change from detection output format to YOLO format

---

Input: the image and its corresponding text File

Output: the text file with YOLO bounding box format

Procedure:

1. classes ← an array of categories list ['pants', 'mid-length dress', 'hoodie', 'jacket', 'mid-length skirt', ‘necklace’, ‘belt’, ‘ring’, ‘earrings’, ‘bracelet’]
2. *H* ← the height of the image
3. *W* ← the width of the image
4. **read** the text file
5. B ← an array of the total number of bounding box values from the text file
6. **IF** the length of B > 0:
7.   **FOR** i = 0 to the length of B:
8.   idx ← the value of the predicted classes of B[i]
9.   *left_x* ← the value of the *left_x* of B[i]
10.   *top_y* ← the value of the *top_y* of B[i]
11.   *w* ← the value of the width of B[i]
12.   *h* ← the value of the height of B[i]
13.   *a* = the index of the idx value in classes []
14. $\hat{x} = (left\_x + w/2) / W$
15. $\hat{y} = (top\_y + h/2) / H$
16. $\widehat{W} = w / W$
17. $\widehat{H} = h / H$

---

boundingBoxYOLO ← **concatenate** (a + “ ” + $\hat{x}$ + “ “+$\hat{y}$ +” “+$\widehat{W}$ +” “+$\widehat{H}$)

write boundingBoxYOLO value in the text file

ENDIF

Algorithm 1 is the pseudocode of pseudo-labeling to change from detection output format to YOLO format. We define "classes" as an array of category lists for "pants", "mid-length dresses", "hoodies", "jackets", "mid-length skirts", “necklaces”, “belts”, “rings”, “earrings”, “bracelets”. After saving the result, we deleted some lines that we did not need to calculate. B is an array of the total number of bounding boxes from the text file because some images have many categories to detect. For every bounding box, we put the values of the predicted class of B[i] to the "idx". "left_x" is for the values of the left_x of B in the i^{th} Bounding box, and top_y is for the values of top_y in ith Bounding Box. w and h are the width and height values of the bounding box in ith iterations shown in Algorithm 1. After that, we identify the actual value of the class that the LabelImg tool already defined by comparing the "classes" array. Furthermore, the conversion process from the detection output format into the YOLOV4 format begins by performing calculations to get the value of $\hat{x}$, $\hat{y}$, $\widehat{W}$ and $\widehat{H}$. These values are then executed using the boundingBoxYOLO function, which is concatenated with a, $\hat{x}$, $\hat{y}$, $\widehat{W}$ and $\widehat{H}$.

After the pseudo-labeling process, we manually annotated only 663 images and obtained a total of 5,772 labeled images of Dataset A and Dataset B. We used image data augmentation techniques to obtain a larger image to improve the accuracy of the model. Only a few Shopee vendors photograph their products professionally in a studio. There is a risk, particularly for those who sell second-hand apparel, that the garments will be photographed with a cluttered background. As a result, it is critical to be able to add a variety of different sorts of objects to the training dataset in order to clearly distinguish the categories. To increase the richness of the experimental dataset, our images were pre-processed in terms of brightening, rotation, mosaic, and CLAHE, and the dataset was augmented.

We chose 0.6 and 1.4 as the lowest and maximum values for brightening the training photos. Three values were chosen from that collection, and three new photos with varying brightness levels were added to the training dataset. This method can be used to imitate the state of clothing under various levels of illumination. To further supplement the image collection, Mosaic data augmentation was used to augment the training images. Mosaic data augmentation combines four training photos in specific ratios to create a single image. This enables the model to acquire the ability to recognize items in a smaller size than usual. Contrast Limited Adaptive Histogram Equalization (CLAHE) is utilized as the final stage to increase the illumination of the garment pattern. The rotation technique is used to improve the illumination of the object pattern during the final step of augmentation. Figure 3.13 shows the sample of training images after image augmentation. Table 3.3 represents the number of images generated by pseudo-labeling and image data augmentation. We labeled only 663 images manually, and 18,203 images were automatically labeled. We trained our proposed FC-YOLOv4 model using obtained labeled images and assessed it by comparing the results to those obtained with YOLOv4 and YOLOv3.

(a) (b) (c) (d)

Figure 3. 13 Samples of training images after image augmentation (a) Brightening, (b) Mosaic, and (c) CLAHE (d) Rotation results

Table 3. 3 The number of images generated by pseudo-labeling and data augmentation

Before Pseudo-labelling (Dataset A)	After Pseudo-labelling (Dataset B)	Rotation	Brightening	Mosaic	CLAHE	Total
1,213	5,772	1,655	17,316	1,074	3,453	30,483

CHAPTER 4

RESEARCH RESULT

This section includes the result of the content and thematic analysis. The performance of the category detection model is compared to that of other approaches in this section. Meanwhile, the outcomes of fashion category prediction using YOLOv4 and YOLOv3 are compared and studied.

4.1 Content Analysis Result

We analyzed a product category using its product photos and category data Figure 4.1 using Google Spread Sheet. If the seller specifies a category for a product that does not match the image of the product, the category is FALSE; if they match, the category is TRUE. We checked each product manually and configured the category error rate. As a result of the seller improperly categorizing the product, 14.1% were placed in the "Dress" category, 8.9% in the "Skirt" category, and less than 5% in the shirt, denim, and trouser category, and 50% in the jumpsuit category.

Figure 4. 1 Content Analysis Using Google Spread Sheet

4.2 Thematic Analysis Result

Thematic Analysis is performed to determine which other existing categories were included in the "other" category. In the "other" category, we investigated the category of each product based on the product image and title (Figure. 4.2). We manually reviewed each product title to see whether it had any other comparable words from the category word or category as shown in Table 4.1. For instance, if the product title contains the words shirt, t-shirt, or polo shirt, the product is classed as a shirt.

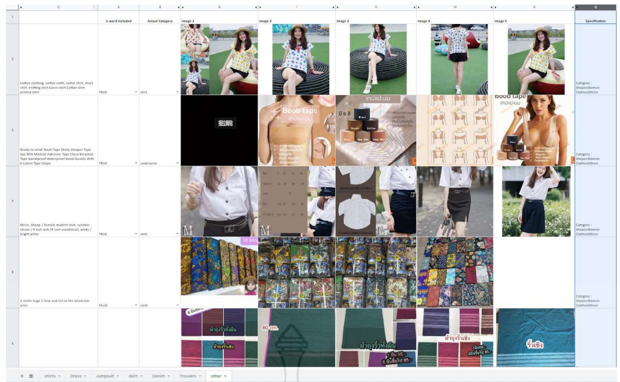


Figure 4. 2 Thematic Analysis Using Google Spread Sheet

Table 4. 1 Table Type Styles

Category	Words
Shirt	“shirt”, “t-shirt”
Dress	“dress”
Jumpsuit	“jumpsuit”
Skirt	“skirt”
Denim	“denim”, “jeans”
Trousers	“trousers”, “pants”
Jackets and coats	“jackets”, “coats”
Underwear	“underwear”, “bikini”
Pajamas	“pajamas”, “nightwear”
set	“set”
cloth	“cloth”

When sellers select a product category, on average, 29% of products are placed in the incorrect category as shown in Figure. 4.3. As you can see, 75.1 percent of products have their own category yet are categorized as "other" (Figure.4.4). This means that the majority of sellers add their products to the "other" category rather than using the Shopee-recommended category list. According to the report, 72.7 percent of product titles contain category words or their synonyms (Figure. 4.5). Thus, when classifying a product, it is critical to keep in mind that the product title is equally significant.

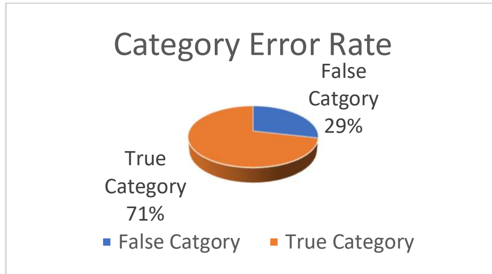


Figure 4. 3 Content Analysis Results

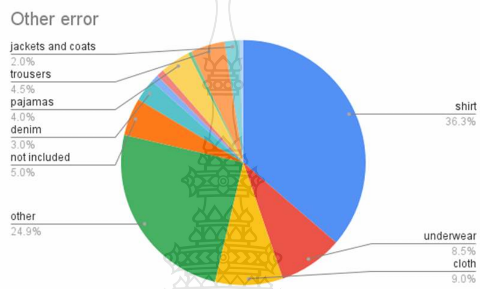


Figure 4. 4 Percentage of Category Included in Other's Category

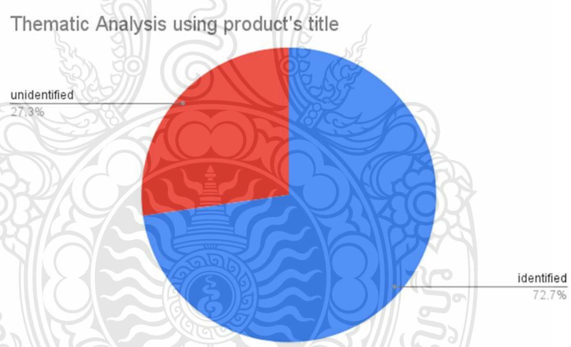


Figure 4. 5 Thematic Analysis Result

4.3. Comparison of Before and After Pseudo-Labeling With Yolov4

In our research, we employed our YOLOv4 model with the CSPDarknet53. The parameters used to initialize the network are listed in Table 4.2. To improve the model's detection accuracy and alter the input required by the Darknet framework, we utilized the size of the input images in our research to 416 x 416. For the purpose of evaluating the training procedure, the batch size was set to 64, resulting in the capture of 64 images every iteration for 8000,9000 training steps. Momentum, initial learning rate, weight decay regularization, and other parameters were the original parameters of the YOLOv4 model. We set the maximum batch size to 10,000 (the number of classes multiplied by 2,000) and increase it in 80 and 90 percent increments. We apply filters based on the (number of classes + 5) * 3 formula. It will process 64 images every iteration during training and transmit 24 subdivisions to the GPU simultaneously.

Table 4. 2 Initialization Parameters of YoloV4 Network

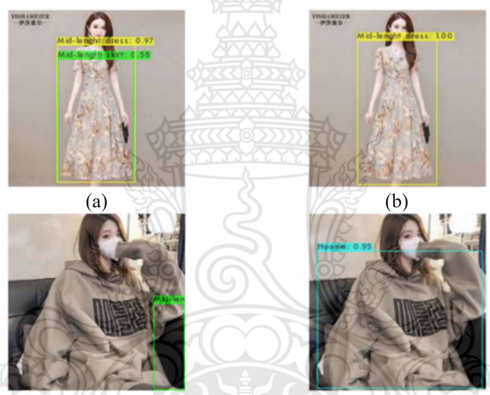
Input image size	Batch	Momentum	Initial learning rate	Decay	Training steps
416 x 416	64	0.949	0.001	0.0005	8000,9000

(Nguyen et al., 2021)studied that the performance of the models was considerably altered when alternate training and validation ratios were applied, demonstrating that a 70:30 ratio of training to testing datasets is optimal for the object detection model. Dataset A was partitioned into 90% for training and 10% for validation during the first training. After training, we predicted dataset B and converted the predicted output to YOLO format. Then, we mixed the labeled dataset A and predicted dataset B for the second training set, divided into 80% for the training set and 10% for the testing and validation sets, respectively.

To evaluate the influence of dataset size on detection outcomes, pictures from all categories were pooled before and after pseudo-labeling to train the model. We computed and assessed the performance of the model on dataset A, which includes Average Precision (AP), True Positive (TP), False Positive (FP), Recall, IOU, and Mean Average Precision (mAP). Table 4.3 shows the performance metrics before pseudo-labeling and after pseudo-labeling. In Class Mid-Length Dresses, Hoodie, Jacket, and Mid-Length Skirt, there are considerable disparities in AP, TP, and FP results. Figure 4.6 (a) and 4.6 (c) are the detection results before pseudo-labeling and Figures 4.6 (b) and 4.6 (d) are the detection results after pseudo-labeling. Before pseudo-labeling, our model incorrectly predicted two identical categories in Figure 4.6 (a), but the bounding box could be predicted successfully. Even the bounding box in Figure 4.6 (c) could not be adjusted appropriately. Following pseudo-labeling, the category and bounding box may be predicted precisely, as seen in Figures 4.6 (b) and 4.6 (d). Overall, YOLOv4 has a good level of accuracy, with a 0.97 mAP after pseudo-labeling. This is because there are fewer datasets before and after pseudo-labeling.

Table 4. 3 Performance Metrics on YOLOv4 Before Pseudo-Labeling and After Pseudo-Labeling

Class	Before pseudo-labelling				After pseudo-labelling			
	AP	TP	FP	mAP@0.5	AP	TP	FP	mAP@0.5
Pants	94.06%	790	212		97.55%	808	54	
Mid-length dress	46.68%	282	58		96.80%	716	42	
Hoodie	50.09%	388	142	0.62	96.53%	882	40	0.97
Jacket	61.75%	270	22		97.19%	512	36	
Mid-length skirt	60.23%	864	354		99.32%	1526	58	



(a) (b) (c) (d)

Figure 4. 6 Detection results of before and after pseudo-labeling

4.4. Comparison of Our FC-YOLOv4 With YOLOv4 and YOLOv3 Models

In this part, we carried out a series of tests using the trained FC-YOLOv4 model and test images to validate the algorithm's performance. We used 80 percent of the combination of datasets from before pseudo-labeling, after pseudo-labeling, Brightening, Mosaic, and CLAHE as training datasets, and a total of 16,276 images were used. We divided our dataset into training and testing subsets to ensure that the images in each category were balanced. For the testing dataset, we used the collection of the rest of 20 percent of our dataset and collection of the images from the DeepFashion2 dataset, second-hand images from Google Image, and Augmented images. When evaluating the model, we also collected clutter backdrop photos that had multiple categories. The suggested model is compared against the YOLOv3 and YOLOv4 two-stage conventional models in order to demonstrate the proposed model's superiority. The primary purpose of this research is that our model aims to be able to categorize the closest category when it detects an image that is not in the trained dataset.

Several criteria, including precision (P), recall (R), F1 score, AP, mAP, and IoU, were utilized to compare the performance of the three models [37]. The accuracy of a set of object detection from the model is evaluated using mAP. The amount of overlap between the predicted bounding box and the underlying truth is measured using mAP, which ranges from 0 to 1, as opposed to IoU, which is employed while calculating mAP. Equation 3-8 depicts the mathematical representation of these six measures.

$$P = \frac{TP}{FP + TP} \quad \text{......................} \ (12)$$

$$R = \frac{TP}{FN + TP} \quad \text{......................} \ (13)$$

$$F1\,score = \frac{2PR}{P + R} \quad \text{......................} \ (14)$$

$$AP = \int_0^1 p(r)dr \quad \text{......................} \ (15)$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad \text{......................} \ (16)$$

$$IoU = \frac{Area\ of\ overlap}{Area\ of\ union} \quad \text{......................} \ (17)$$

The precision-recall curve, or P-R curve, can be generated by plotting the precision ratio against the recall ratio using the precision ratio as the vertical axis and the recall ratio as the horizontal axis. The precision-recall curves for our three models are depicted in Figure 4.7. While YOLOv4 is more accurate at lower IOU thresholds, our FC-YOLO v4 is more efficient at higher IOU thresholds. As shown in Table 4.4, after data augmentation, our proposed FC-YOLOv4 model gets 0.007 percent and 40.2 percent higher than the original YOLOv4 and YOLOv3 models with nearly the same detection time.

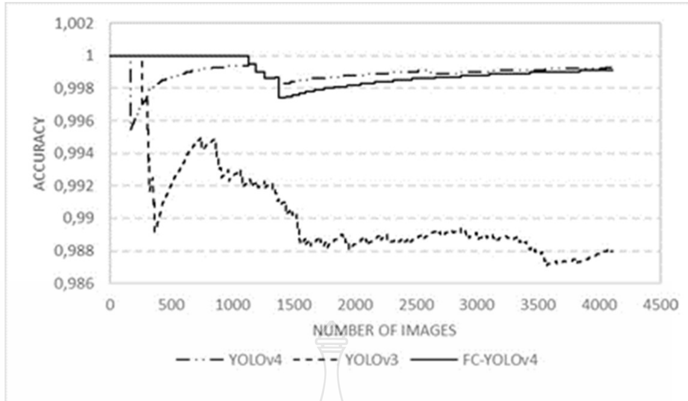


Figure 4. 7 P-R curve

Table 4. 4 Comparison of parameters, detection time, accuracy, and size of three models

	Total parameters	Detection time	Accuracy(mAp)	size
FC-YOLOv4	63,959,226	33 milliseconds	95.87%	266MB
YOLOv4	63,959,226	32 milliseconds	95.80%	244MB
YOLOv3	61,545,274	32 milliseconds	55.67%	234MB

(a) (b) (c)

(d) (e) (f)

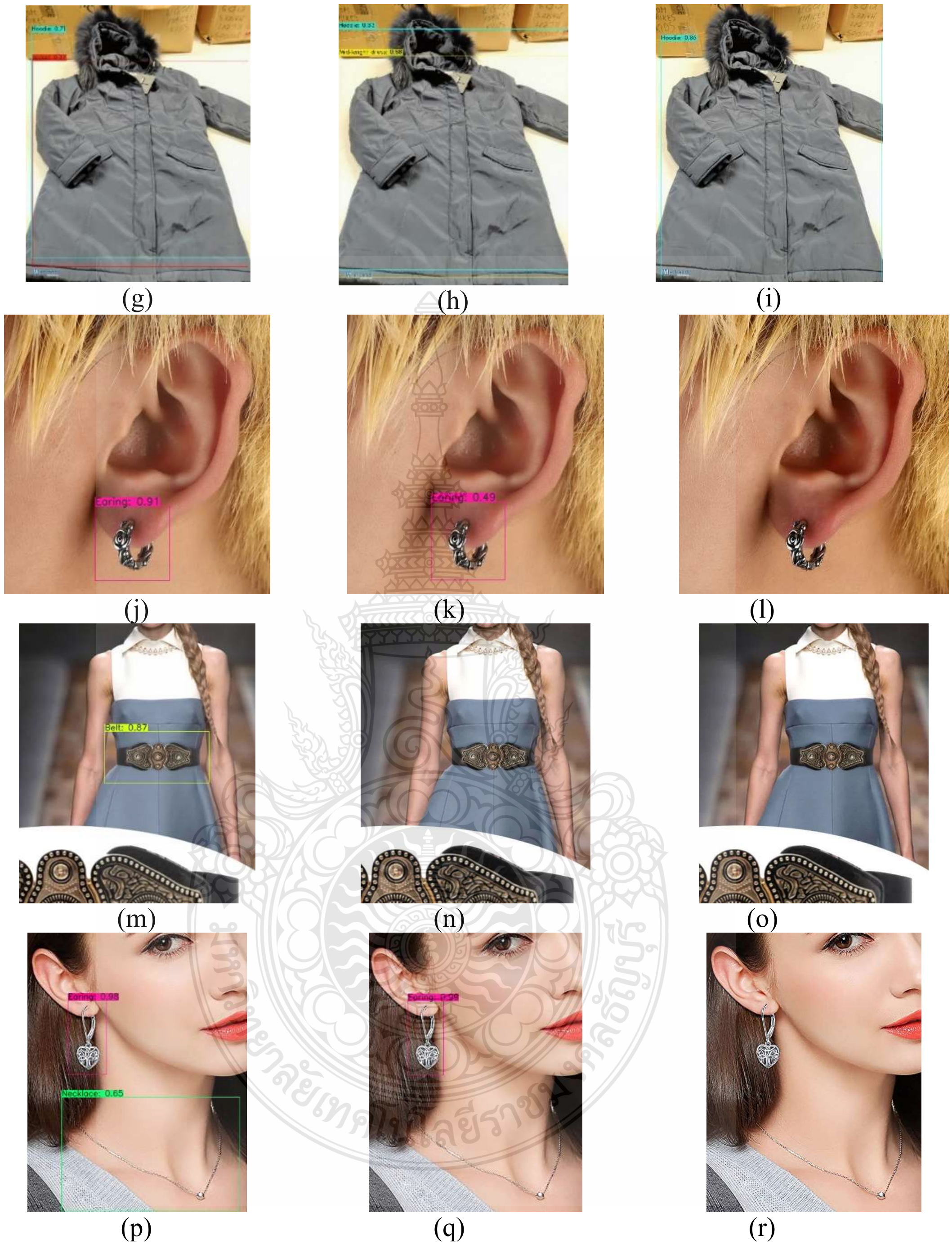


Figure 4. 8 Detection results of YOLOv3, YOLOv4, and our proposed FC-YOLOv4

The detection results of our suggested model were compared to that of the YOLOv3 and Yolov4 conventional models. We evaluated photos with many categories to verify the model’s correctness. The detection results for the YOLOv3 model are described in Figures 4.8 (a), 4.8 (d), and 4.8 (g). The detection findings for YOLOv4 are illustrated in Figures 4.8 (b), 4.8 (e), and 4.8 (h), whereas the detection findings for our FC-YOLOv4 model are shown in Figures 4.8 (c), 4.8 (f), and 4.8 (i). In Figures 4.8 (a), 4.8 (b), and 4.8 (c), we discovered that our model could classify the pattern as a “skirt” and can recognize two bounding box skirts even when the training dataset has a different pattern and part of the skirt is covered by the coat. When we examine the findings of the three models in Figures 4.8 (d), 4.8 (e), and 4.8 (f), we can see that even though one of the jackets was not photographed from the front view, only the FC-YOLOv4 model correctly detects it. In Figures 4.8 (j), 4.8 (k), and 4.8 (l), our FC-YOLOv4 model correctly classifies items as “hoodies,” while the other two models’ bounding boxes overlap.

Figure 4.9 depicts the outcomes of our suggested FC-YOLOv4 model’s detection. As displayed in Figures 4.9(a), 4.9(b), 4.9(c), and 4.9(d), our proposed model is capable of detecting overlapping clothing with varying hue/saturation levels. As illustrated in Figures 4.9(e), 4.9(f), and 4.9(g), our model can recognize even a white outfit with a high brightness level. Figures 4.9(h), 4.9(i), 4.9(j), and 4.9(k) illustrate that our suggested model can also be used to identify secondhand clothing with a cluttered background. As depicted in Figures 4.9(l), 4.9(m), and 4.9(n), our proposed model can accurately detect rotated images. Table 4.5 represents the number of true positive results with different thresholds. Our FC-YOLOv4 model has more True Positive numbers than another two models.

(a) (b) (c) (d)

Figure 4. 9 Detection results of our proposed FC-YOLOv4

Table 4. 5 Number of true positive results with different thresholds

IOU	FC-YOLOv4	YOLOv4	YOLOv3
0	8,111	8,091	6,119
0.1	7,745	7,766	4,263
0.2	7,583	7,575	3,674
0.3	7,444	7,440	3,301
0.4	7,307	7,292	3,008
0.5	7,132	7,127	2,733
0.6	6,957	6,961	2,442
0.7	6,737	6,705	2,155
0.8	6,403	6,358	1,794
0.9	5,687	5,613	1,324

In each class, the YOLOv3, YOLOv4, and FC-YOLOv4 models are compared using a summary of the results. It was observed that after 20000 epochs, FC-YOLOv4's mAP% was better than YOLOv4 and YOLOv3. As seen in Table 3, the FC-YOLOv4 model yielded superior results for every statistic. In the validation set, the FC-YOLOv4 model attained the highest mAP percentage of 99.84 percent. Compared to the intended model, YOLOv3 reached 93.74 percent, and YOLOv4 achieved 99.81 percent, with margins of 6.1% and 0.3%. The P and R values for YOLOv3 were 0.96 and 0.74, while YOLOv4 and FC-YOLOv4 had ratings of 0.99 and 0.99, respectively. Therefore, FC-YOLOv4 and YOLOv4 performed better than YOLOv3 by 3.03 percent for P and by 25.25 percent for R. F1 scores were determined as 0.83 for YOLOv3, 0.99 for YOLOv4, and 0.99 for FC-YOLOv4. FC-YOLOv4 and YOLOv4, therefore, performed 16.16 percent better than YOLOv3. FC-YOLOv4 is also superior to YOLOv4 and YOLOv3 by 12.49 percent and 1.23 percent, respectively, in terms of IoU.

Table 4.6. Evaluation Matrices

Model	Iterations	TP	FP	FN	P	R	F1-score	mAP (%)	IoU (%)
YOLOv3	20000	21649	902	7713	0.96	0.74	0.83	93.74	77.44
YOLOv4		29116	259	246	0.99	0.99	0.99	99.81	87.40
FC-YOLOv4		29093	179	269	0.99	0.99	0.99	99.84	88.49

4.5. Evaluation of the approach in mobile deployment

The TensorFlow Lite technique modifies the configuration files and weights of YOLOv3, YOLOv4, and FC-YOLOv4 to produce models with lower memory footprints. Table 4 compares the parameter of each model. YOLOv3 has the smallest model size, YOLOv4 the second biggest, and FC-YOLOv4 the largest. On the basis of the regular and lite sizes of the three models, one may deduce that the average size has decreased by 50 percent. In this investigation, the smaller model will be loaded into memory more quickly. YOLOv3, with a model size of 117 MB, loaded in 25,268 milliseconds. YOLOv4, with a model size of 122 MB, loaded in 31,865 milliseconds. And FC-YOLOv4, with a model size of 151 MB, has a loading time of 34.06 milliseconds. This may be due to the smaller network size of YOLOv3 and YOLOv4 relative to FC-YOLOv4, which enables YOLOv3, and YOLOv4 to learn quickly while FC-YOLOv4 learns precisely.

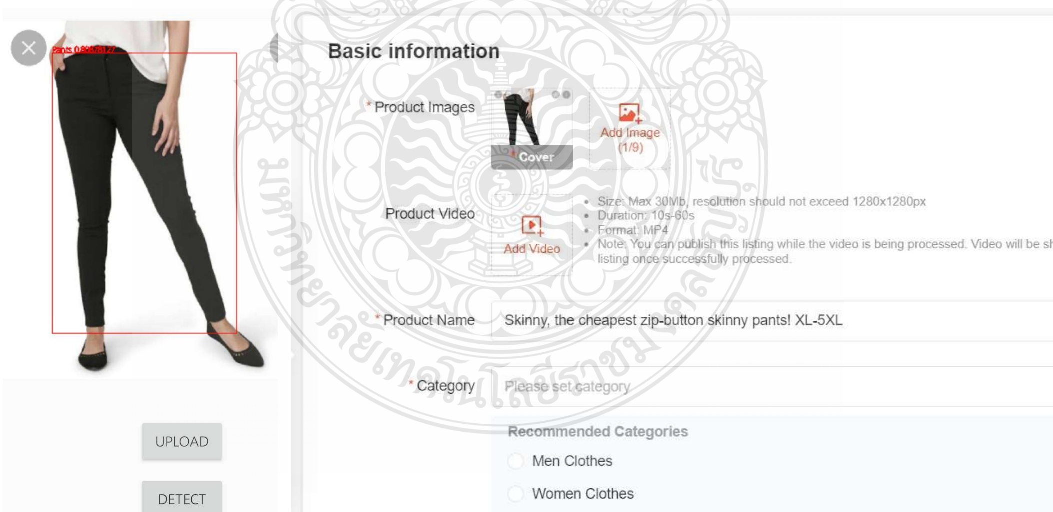
Object recognition and image processing are examples of computationally complex jobs that can result in a higher average RAM consumption. This subsection examines the RAM used in the object recognition algorithm to recognize numerous images. According to Table 2, YOLOv3, YOLOv4, and FC-YOLOv4 use an average of 7.4 MB, 7.2 MB, and 1.1 MB of RAM, respectively. Comparatively, the greatest RAM consumption is 769MB, 1.3GB, and 576MB. Therefore, we may infer that FC-YOLOv4 performed the best.

Table 4. Evaluation of Mobile Application Deployment

Parameters	YOLOv3 Lite	YOLOv4 Lite	FC-YOLOv4 Lite
Normal Model Size	235.1 MB	244.3 MB	302.4 MB
Lite Model Size	117 MB	122 MB	151 MB
Application Size	138 MB	137 MB	168 MB
Loading Time	25.268 ms	31.865 ms	34.06 ms
Average RAM usage	7.4 MB	7.2 MB	1.1 MB
Maximum RAM usage	769 MB	1.3 GB	567 MB

4.6. Evaluation of the approach in a real scenario

The final evaluation of this study compares the results of FC-YOLOv4 Lite detection on a mobile device to the Shopee web application. We collected some of the products with the wrong categories from the Shopee Thailand Website. Figure 6 depicts one of the test results for a real-world scenario in which FC-YOLOv4 accurately recognizes the Pants category while the Shopee recommendation system recognizes Men's and Women's clothing.



(a) (b)

Figure 6. Real Scenario Comparison. (a) FC-YOLOv4 Mobile Application; (b) Shopee Web Application

In our research of how popular object detectors such as YOLOv4 can be modified to detect objects, we identified architectural modifications that yield a significant performance boost over the original model at a relatively cheap cost, as the new model retains speed. The scenario in which we applied the suggested approach, namely autonomous racing, might benefit significantly from such enhancements. We not only enhanced the performance of the base model greatly in this study but also discovered some specialized strategies that may be applied to various applications requiring the detection of objects. Consequently, a modified model with pseudo-labeling outperforms a YOLOv4 class model while keeping a detection speed suitable for fashion applications.

Finally, while this study reveals a sizeable empirical benefit of the recommended architectural adjustments, the study's consistency and generalizability may and should be further studied. For instance, the analysis would benefit significantly from more testing with diverse data sets and the issues that may occur when detecting firm brands, for example. While we have established the utility of the various strategies presented, they can only be improved and better understood via application to several diverse situations and places. This would be a huge step in developing a more robust system for detecting fashion objects. Additionally, several additional paths and strategies apply to this issue but were not studied; nonetheless, they will remain the focus of future research.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

When it comes to purchasing products and services online, customers now have more options. The outcomes of the research are the proportion of times the automatic category recommendation method causes sellers to select the erroneous category for their product. Additionally, this study summarizes consumer confidence in the product category prior to making a purchase. Raising customer expectations and expediting the consumer purchasing decision-making process benefits both suppliers and buyers by increasing profits.

We collected a currently collected small dataset from Shopee E-commerce Thailand. Furthermore, we introduced a semi-supervised technique in our research for predicting fashion product categories using minimum labeled data and a huge amount of unlabeled data. In contrast, the majority of previous work has concentrated on the performance of object identification models. We used FC-YOLOv4 to detect the product category in Shopee's fashion category. This method simplifies the process of selecting a category for sellers and enables a more targeted recommendation system. And then, we compared our FC-YOLOv4 model with YOLOv4 and YOLOv3 models by using secondhand clothing, DeepFashion2, and augmented images. Our research established that model training accuracy is much greater after pseudo-labeling than before pseudo-labeling. When the performance of the three models is evaluated, our YOLOv4 model detects more categories inside an image than the other two models. All verified measures, such as recall, accuracy, and IOU, improved in value. Using pseudo-labeling, we can minimize the amount of manual labeling necessary for data training and enhance model accuracy. Because pseudo-labeling is retrained from the model and re-labels the images. This study focuses not only on model building and testing but also on the deployment of mobile applications. In addition, the deployed model is evaluated by comparing it to the official Shopee website. The main challenge with this study is that labeling mistakes may arise if the initially labeled dataset is too small or the model's detection performance is inadequate.

Additionally, we may collect information about client behavior in the future to help us improve the overall quality of Shopee's e-commerce platform. We will do more in-depth comparative analyses of various state-of-the-art machine learning algorithms to significantly improve the performance of the product image categorization architecture described in this research. In future work, we will investigate a machine-learning method for classifying products based on their title and images. We would want to suggest that instead of detecting each product individually, we categorize them all at once into their respective categories. Additionally, we would like to focus on picture categorization that considers the hierarchical structure of garment categories.

LIST OF BIBLIOGRAPHY

Agus, A. A., Yudoko, G., Mulyono, N., & Imaniya, T. (2021). E-Commerce Performance, Digital Marketing Capability and Supply Chain Capability within E-Commerce Platform: Longitudinal Study Before and After COVID-19. *International Journal of Technology*, *12*(2), 360–370. https://doi.org/10.14716/ijtech.v12i2.4122

Amornkitvikai, Y., & Lee, C. (2020). *Determinants of E-Commerce Adoption and Utilisation by SMEs in Thailand*.

Arulprakash, E., & Aruldoss, M. (2021). A study on generic object detection with emphasis on future research directions. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2021.08.001

Bauer, G. R., & Scheim, A. I. (2019a, April 1). Advancing quantitative intersectionality research methods: Intracategorical and intercategorical approaches to shared and differential constructs. *Social Science and Medicine*, Vol. 226, pp. 260–262. Elsevier Ltd. https://doi.org/10.1016/j.socscimed.2019.03.018

Bauer, G. R., & Scheim, A. I. (2019b, April 1). Advancing quantitative intersectionality research methods: Intracategorical and intercategorical approaches to shared and differential constructs. *Social Science and Medicine*, Vol. 226, pp. 260–262. Elsevier Ltd. https://doi.org/10.1016/j.socscimed.2019.03.018

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. Retrieved from http://arxiv.org/abs/2004.10934

Bossard, L., Dantone, M., Leistner, C., Wengert, C., Quack, T., & Gool, L. van. (n.d.-a). *Apparel Classification with Style*.

Bossard, L., Dantone, M., Leistner, C., Wengert, C., Quack, T., & Gool, L. van. (n.d.-b). *Apparel Classification with Style*.

Boysen, N., de Koster, R., & Weidinger, F. (2019, September 1). Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*, Vol. 277, pp. 396–411. Elsevier B.V. https://doi.org/10.1016/j.ejor.2018.08.023

Chai, E., Ta, L., Ma, Z., & Zhi, M. (2021). ERF-YOLO: A YOLO algorithm compatible with fewer parameters and higher accuracy. *Image and Vision Computing*, *116*. https://doi.org/10.1016/j.imavis.2021.104317

Chaparro-Peláez, J., Hernández-García, Á., & Urueña-López, A. (2015). The role of emotions and trust in service recovery in business-to-consumer electronic commerce. *Journal of*

Theoretical and Applied Electronic Commerce Research, *10*(2), 77–90. https://doi.org/10.4067/S0718-18762015000200006

Chen, H., Gallagher, A., & Girod, B. (n.d.). *Describing Clothing by Semantic Attributes*.

Chen, M., Qin, Y., Qi, L., & Sun, Y. (n.d.). *Improving Fashion Landmark Detection by Dual Attention Feature Enhancement*. Retrieved from https://codalab.

Choshin, M., & Ghaffari, A. (2017). An investigation of the impact of effective factors on the success of e-commerce in small- and medium-sized companies. *Computers in Human Behavior*, *66*, 67–74. https://doi.org/10.1016/j.chb.2016.09.026

David Tinambunan, I. (2019). Comparative Analysis of Shopee and Lazada Web Service (Study on Shopee and Lazada Users in Jakarta City). *International Research Journal of Advanced Engineering and Science*, *4*(3), 385–388.

de Souza Inácio, A., & Lopes, H. S. (2020). Epynet: Efficient pyramidal network for clothing segmentation. *IEEE Access*, *8*, 187882–187892. https://doi.org/10.1109/ACCESS.2020.3030859

Donati, L., Iotti, E., Mordonini, G., & Prati, A. (2019). Fashion product classification through deep learning and computer vision. *Applied Sciences (Switzerland)*, *9*(7). https://doi.org/10.3390/app9071385

Erlingsson, C., & Brysiewicz, P. (2017, September 1). A hands-on guide to doing content analysis. *African Journal of Emergency Medicine*, Vol. 7, pp. 93–99. African Federation for Emergency Medicine. https://doi.org/10.1016/j.afjem.2017.08.001

Fu, L., Duan, J., Zou, X., Lin, J., Zhao, L., Li, J., & Yang, Z. (2020). Fast and accurate detection of banana fruits in complex background orchards. *IEEE Access*, *8*, 196835–196846. https://doi.org/10.1109/ACCESS.2020.3029215

Fujii, K., & Kawamoto, K. (2021). Generative and self-supervised domain adaptation for one-stage object detection. *Array*, *11*, 100071. https://doi.org/10.1016/j.array.2021.100071

Girshick, R. (2015). *Fast R-CNN*. Retrieved from http://arxiv.org/abs/1504.08083

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 580–587. IEEE Computer Society. https://doi.org/10.1109/CVPR.2014.81

Happ, E., Scholl-Grissemann, U., Peters, M., & Schnitzer, M. (2021). Insights into customer experience in sports retail stores. *International Journal of Sports Marketing and Sponsorship*, *22*(2), 312–329. https://doi.org/10.1108/IJSMS-12-2019-0137

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-October*, 2980–2988. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2014). *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*. https://doi.org/10.1007/978-3-319-10578-9_23

He, K., Zhang, X., Ren, S., & Sun, J. (2015a). *Deep Residual Learning for Image Recognition*. Retrieved from http://arxiv.org/abs/1512.03385

He, K., Zhang, X., Ren, S., & Sun, J. (2015b). *Deep Residual Learning for Image Recognition*. Retrieved from http://arxiv.org/abs/1512.03385

Hsieh, Y. H., Wu, S. H., Chen, L. P., & Yang, P. C. (2017). Constructing hierarchical product categories for E-commerce by word embedding and clustering. *Proceedings - 2017 IEEE International Conference on Information Reuse and Integration, IRI 2017*, *2017-January*, 397–402. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IRI.2017.81

Huang, E., Su, Z., Zhou, F., & Wang, R. (2020). Learning rebalanced human parsing model from imbalanced datasets. *Image and Vision Computing*, *99*. https://doi.org/10.1016/j.imavis.2020.103928

Hwangbo, H., Kim, Y. S., & Cha, K. J. (2018). Recommendation system development for fashion retail e-commerce. *Electronic Commerce Research and Applications*, *28*, 94–101. https://doi.org/10.1016/j.elerap.2018.01.012

International JOurnal of Sales, Retailing & Marketing. (n.d.).

Jia, M., Shi, M., Sirotenko, M., Cui, Y., Cardie, C., Hariharan, B., … Belongie, S. (2020). *Fashionpedia: Ontology, Segmentation, and an Attribute Localization Dataset*. Retrieved from http://arxiv.org/abs/2004.12276

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, *7*, 128837–128868. https://doi.org/10.1109/ACCESS.2019.2939201

Khalaf, R., & Akbulut, Y. (2021). Smart Arms Detection System Using YOLO Algorithm and OpenCV Libraries. In *Turkish Journal of Science & Technology* (Vol. 16).

Kim, H. J., Lee, D. H., Niaz, A., Kim, C. Y., Memon, A. A., & Choi, K. N. (2021). Multiple-Clothing Detection and Fashion Landmark Estimation Using a Single-Stage Detector. *IEEE Access*, *9*, 11694–11704. https://doi.org/10.1109/ACCESS.2021.3051424

Kumar Addagarla, S. (2022). Single Stage Deep Transfer Learning Model for Apparel Detection and Classification for E-Commerce. *International Journal of Electronic Commerce Studies*, *13*(1), 69–92. https://doi.org/10.7903/ijecs.1953

Law, H., & Deng, J. (2018). *CornerNet: Detecting Objects as Paired Keypoints*. Retrieved from http://arxiv.org/abs/1808.01244

Lawrence, J. E., & Tar, U. A. (2010). Barriers to ecommerce in developing countries. *Information, Society and Justice*, *3*(1), 23–35.

Lee, C. H., & Lin, C. W. (2021). A two-phase fashion apparel detection method based on yolov4. *Applied Sciences (Switzerland)*, *11*(9). https://doi.org/10.3390/app11093782

Leibe, B., Matas, J., Sebe, N., & Welling, M. (Eds.). (2016). *Computer Vision – ECCV 2016*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0

Lester, J. N., Cho, Y., & Lochmiller, C. R. (2020). Learning to Do Qualitative Data Analysis: A Starting Point. *Human Resource Development Review*, *19*(1), 94–106. https://doi.org/10.1177/1534484320903890

Li, J., Zhao, J., Wei, Y., Lang, C., Li, Y., Sim, T., … Feng, J. (2017). *Multiple-Human Parsing in the Wild*. Retrieved from http://arxiv.org/abs/1705.07206

Li, N., Cheng, B., & Zhang, J. (2022). A Cascade Model with Prior Knowledge for Bone Age Assessment. *Applied Sciences*, *12*(15), 7371. https://doi.org/10.3390/app12157371

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). *Focal Loss for Dense Object Detection*. Retrieved from http://arxiv.org/abs/1708.02002

Liu, K. H., Liu, T. J., & Wang, F. (2021). Cbl: A clothing brand logo dataset and a new method for clothing brand recognition. *European Signal Processing Conference*, *2021-January*, 655–659. European Signal Processing Conference, EUSIPCO. https://doi.org/10.23919/Eusipco47968.2020.9287647

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). *Path Aggregation Network for Instance Segmentation*. Retrieved from http://arxiv.org/abs/1803.01534

Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). *DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations*. Retrieved from https://www.google.com/imghp

Lohia -Guise, A. (n.d.). *Bibliometric Analysis of One-stage and Two-stage Object Detection*.

Malhotra, C. (2021). *School of Management Studies*.

Mameli, M., Paolanti, M., Pietrini, R., Pazzaglia, G., Frontoni, E., & Zingaretti, P. (2022). Deep Learning Approaches for Fashion Knowledge Extraction from Social Media: A Review.

IEEE Access, Vol. 10, pp. 1545–1576. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2021.3137893

Mitchell, T. (1997). *Machine Learning, McGraw-Hill, New York*.

Mohd Dahlan, M. (2019). ONLINE SHOPPING: HOW THE BUSINESS PRODUCT CATEGORY AFFECTS THE CONSUMER TRUST TO PURCHASE ONLINE IN MALAYSIA. In *International Journal of Management and Applied Science*. Retrieved from http://iraj.in

Nguyen, Q. H., Ly, H. B., Ho, L. S., Al-Ansari, N., van Le, H., Tran, V. Q., … Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, *2021*. https://doi.org/10.1155/2021/4832864

Oyewole, S. A., & Olugbara, O. O. (2018). Product image classification using Eigen Colour feature with ensemble machine learning. *Egyptian Informatics Journal*, *19*(2), 83–100. https://doi.org/10.1016/j.eij.2017.10.002

Queirós, A., Faria, D., & Almeida, F. (n.d.-a). *European Journal of Education Studies STRENGTHS AND LIMITATIONS OF QUALITATIVE AND QUANTITATIVE RESEARCH METHODS*. https://doi.org/10.5281/zenodo.887089

Queirós, A., Faria, D., & Almeida, F. (n.d.-b). *European Journal of Education Studies STRENGTHS AND LIMITATIONS OF QUALITATIVE AND QUANTITATIVE RESEARCH METHODS*. https://doi.org/10.5281/zenodo.887089

Ranjan, A., Misra, M., & Yadav, J. (n.d.). *Online Shopping Behavior During COVID 19 Pandemic: an Indian Perspective*. Retrieved from https://ssrn.com/abstract=3874348

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015a). *You Only Look Once: Unified, Real-Time Object Detection*. Retrieved from http://arxiv.org/abs/1506.02640

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015b). *You Only Look Once: Unified, Real-Time Object Detection*. Retrieved from http://arxiv.org/abs/1506.02640

Redmon, J., & Farhadi, A. (2016). *YOLO9000: Better, Faster, Stronger*. Retrieved from http://arxiv.org/abs/1612.08242

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. Retrieved from http://arxiv.org/abs/1804.02767

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks*. Retrieved from http://arxiv.org/abs/1312.6229

Sharma, P. (2020). *Futuristic growth analysis of e-commerce companies*.

Sichuan Institute of Electronics, & Institute of Electrical and Electronics Engineers. (n.d.). *2018 IEEE 4th International Conference on Computer and Communications (ICCC) : December 7-10, 2018, Chengdu, China*.

Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Retrieved from http://arxiv.org/abs/1409.1556

Stuart, J. , & N. P. (2010). *Artificial Intelligence: a Modern Approach Prentice-Hall. In A Simon & Schuster Company Englewood Cliffs*.

Tan, L., Li, M. Y., & Kok, S. (2020). E-Commerce Product Categorization via Machine Translation. *ACM Transactions on Management Information Systems*, *11*(3). Association for Computing Machinery. https://doi.org/10.1145/3382189

Tan, M., & Le, Q. v. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. Retrieved from http://arxiv.org/abs/1905.11946

The History Began from AlexNet A comprehensive Survey on Deep Learning Approaches. (n.d.).

Tiwari, G., Bhatnagar, B. L., Tung, T., & Pons-Moll, G. (2020). *SIZER: A Dataset and Model for Parsing 3D Clothing and Learning Size Sensitive 3D Clothing*. Retrieved from http://arxiv.org/abs/2007.11610

Tzutalin. (2015). LabelImg.

Wang, C.-Y., Liao, H.-Y. M., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., & Hsieh, J.-W. (2019). *CSPNet: A New Backbone that can Enhance Learning Capability of CNN*. Retrieved from http://arxiv.org/abs/1911.11929

Wang, W., Xu, Y., Shen, J., & Zhu, S.-C. (n.d.). *Attentive Fashion Grammar Network for Fashion Landmark Detection and Clothing Category Classification*.

Wu, H., Gao, Y., Guo, X., Al-Halah, Z., Rennie, S., Grauman, K., & Feris, R. (2019). *Fashion IQ: A New Dataset Towards Retrieving Images by Natural Language Feedback*. Retrieved from http://arxiv.org/abs/1905.12794

Wu, H., Gao, Y., Guo, X., Al-Halah, Z., Rennie, S., Grauman, K., & Feris, R. (2021). *Fashion IQ: A New Dataset Towards Retrieving Images by Natural Language Feedback*. Retrieved from https://github.com/XiaoxiaoGuo/fashion-iq

Xiao, H., Rasul, K., & Vollgraf, R. (2017). *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Retrieved from http://arxiv.org/abs/1708.07747

Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S., & Lan, X. (2020). A review of object detection based on deep learning. *Multimedia Tools and Applications*, *79*(33–34), 23729–23791. https://doi.org/10.1007/s11042-020-08976-6

Yamaguchi, K., Hadi, M., Luis, K., Ortiz, E., & Berg, T. L. (n.d.). *Parsing Clothing in Fashion Photographs*.

Yamin Thwe. (2022). Shopee Image Dataset(Thailand). *IEEE Dataport*.

Yang, X., Zhang, H., Jin, D., Liu, Y., Wu, C.-H., Tan, J., … Wang, X. (2020). *Fashion Captioning: Towards Generating Accurate Descriptions with Semantic Rewards*. Retrieved from http://arxiv.org/abs/2008.02693

Zhang, X., Chen, Y., Zhu, B., Wang, J., & Tang, M. (2020). Semantic-spatial fusion network for human parsing. *Neurocomputing*, *402*, 375–383. https://doi.org/10.1016/j.neucom.2020.03.096

Zhou, T., Qi, S., Wang, W., Shen, J., & Zhu, S. C. (2022). Cascaded Parsing of Human-Object Interaction Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(6), 2827–2840. https://doi.org/10.1109/TPAMI.2021.3049156

Zhou, T., Wang, W., Liu, S., Yang, Y., & Gool, L. van. (n.d.). *Differentiable Multi-Granularity Human Representation Learning for Instance-Aware Human Semantic Parsing*. Retrieved from https://github.com/tfzhou/MG-HumanParsing

APPENDICES

APPENDIX A.
SOURCE CODE

In [1]:
!nvidia-smi

```
Wed Apr  6 01:12:18 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   33C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

In [2]:
```
from google.colab import drive
drive.mount('/content/gdrive')
!ln -s /content/gdrive/My\ Drive/ /mydrive
```

```
Mounted at /content/gdrive
```

Helper Functions

In [3]:
```
def imShow(path):
  import cv2
  import matplotlib.pyplot as plt
  %matplotlib inline
  image = cv2.imread(path)
  height, width = image.shape[:2]
  resized_image = cv2.resize(image,(3*width, 3*height), interpolation = cv2.INTER_C
UBIC)
  fig = plt.gcf()
  fig.set_size_inches(18, 10)
  plt.axis("off")
  plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
  plt.show()


def upload():
  from google.colab import files
  uploaded = files.upload()
  for name, data in uploaded.items():
```

```
    with open(name, 'wb') as f:
      f.write(data)
      print ('saved file', name)

def download(path):
  from google.colab import files
  files.download(path)

def get_number_of_elements(list):
    count = 0
    for element in list:
        count += 1
    return count

def string_len_in_file(fname):
    with open(fname) as f:
        for i, l in enumerate(f):
            pass
    return i + 1

def get_file_names(path):
  from os import walk
  filenames = next(walk(path), (None, None, []))[2]
  return filenames
```

Data Augmentation for Before Pseudo Dataset

In [ ]:

```
# Brighting image augmentation
from numpy import expand_dims
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.image as mpimg

for img in words:
        prefix = img.split("/")[6].split(".")[0]
        ext = img.split("/")[6].split(".")[1]
        image = load_img(img)
        data = img_to_array(image)
        samples = expand_dims(data,0)
        datagen = ImageDataGenerator(brightness_range=[0.6,1.4])
        it = datagen.flow(samples, batch_size=1)
        for i in range(3):
                batch = it.next()
                fimg = batch[0].astype('uint8')
                path = "/content/gdrive/MyDrive/After Augmentation/Brighting_pseudo/B" +
```

```
prefix + str(i) + "." + ext
            mpimg.imsave(path, fimg)
```

In [ ]:

```
# copying txt files
import shutil

for txt in words:
   prefix = txt.split("/")[6].split(".")[0]
   src = "/content/gdrive/MyDrive/Before Augmentation/TestingDataset(pseudo)/" + pre
fix + ".txt"
   for i in range(3):
      dst = "/content/gdrive/MyDrive/After Augmentation/Brighting_pseudo/B" + prefix
+ str(i) + ".txt"
      shutil.copyfile(src, dst)
```

In [ ]:

```
# example of random rotation 90 degree image augmentation
from numpy import expand_dims
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.image as mpimg

for img in words:
      prefix = img.split("/")[6].split(".")[0]
      ext = img.split("/")[6].split(".")[1]
      image = load_img(img)
      data = img_to_array(image)
      samples = expand_dims(data,0)
      datagen = ImageDataGenerator(rotation_range=90)
      it = datagen.flow(samples, batch_size=1)
      for i in range(1):
            batch = it.next()
            fimg = batch[0].astype('uint8')
            path = "/content/gdrive/MyDrive/After Augmentation/Rotation_pseudo/R90"
+ prefix + str(i) + "." + ext
            mpimg.imsave(path, fimg)

for img in words:
      prefix = img.split("/")[6].split(".")[0]
      ext = img.split("/")[6].split(".")[1]
      image = load_img(img)
      data = img_to_array(image)
      samples = expand_dims(data,0)
      # 90 , 180 , mirror
```

```
        datagen = ImageDataGenerator(rotation_range=180)
        it = datagen.flow(samples, batch_size=1)
        for i in range(1):
                batch = it.next()
                fimg = batch[0].astype('uint8')
                path = "/content/gdrive/MyDrive/After Augmentation/Rotation_pseudo/R180
" + prefix + str(i) + "." + ext
                mpimg.imsave(path, fimg)

In [ ]:
import shutil

for txt in words:
    prefix = txt.split("/")[6].split(".")[0]
    src = "/content/gdrive/MyDrive/Before Augmentation/TestingDataset(pseudo)/" + pre
fix + ".txt"
    for i in range(1):
        dst_90 = "/content/gdrive/MyDrive/After Augmentation/Rotation_pseudo/R90" +
prefix + str(i) + ".txt"
        shutil.copyfile(src, dst_90)
        dst_180 = "/content/gdrive/MyDrive/After Augmentation/Rotation_pseudo/R180"
+ prefix + str(i) + ".txt"
        shutil.copyfile(src, dst_180)

# Mosaic

In [ ]:
import random
import cv2
import os
import glob
import numpy as np
from PIL import Image

OUTPUT_SIZE = (512, 512)  # Height, Width
SCALE_RANGE = (0.3, 0.7)
FILTER_TINY_SCALE = 1 / 50  # if height or width lower than this scale, drop it.

DIR = "/content/gdrive/MyDrive/Before Augmentation/TestingDataset(pseudo)"
category_name = ['Pants','Mid-lenght dress', 'Hoodie', 'Jacket', 'Mid-length skirt']

def get_dataset():
  img_paths = []
  annos = []

  for img_path in words:
    txt = img_path.split(".")[0] + ".txt"
```

```
    with open(txt, 'r') as f:
      num_obj = int(string_len_in_file(f.name))
      img = cv2.imread(img_path)
      img_height, img_width, _ = img.shape

      boxes = []
      for _ in range(num_obj):
        obj = f.readline().rstrip().split(' ')
        obj = [float(elm) for elm in obj]
        obj[0] = int(obj[0])

        xmin = max(obj[1], 0)
        ymin = max(obj[2], 0)
        xmax = min(obj[3], img_width)
        ymax = min(obj[4], img_height)

        boxes.append([obj[0], xmin, ymin, xmax, ymax])

      if not boxes:
          continue

    img_paths.append(img_path)
    annos.append(boxes)
  return img_paths, annos


def mosaic(all_img_list, all_annos, idxs, output_size, scale_range, filter_scale=0.):
    output_img = np.zeros([output_size[0], output_size[1], 3], dtype=np.uint8)
    scale_x = scale_range[0] + random.random() * (scale_range[1] - scale_range[0])
    scale_y = scale_range[0] + random.random() * (scale_range[1] - scale_range[0])
    divid_point_x = int(scale_x * output_size[1])
    divid_point_y = int(scale_y * output_size[0])

    new_anno = []
    for i, idx in enumerate(idxs):
        path = all_img_list[idx]
        img_annos = all_annos[idx]

        img = cv2.imread(path)
        if i == 0:  # top-left
            img = cv2.resize(img, (divid_point_x, divid_point_y))
            output_img[:divid_point_y, :divid_point_x, :] = img
            for bbox in img_annos:

                # As YOLO annotations have different centers from the image, this is how the
bbox coordinates are calculated
```

```
                xmin = bbox[1] - bbox[3]*0.5
                ymin = bbox[2] - bbox[4]*0.5
                xmax = bbox[1] + bbox[3]*0.5
                ymax = bbox[2] + bbox[4]*0.5

                xmin *= scale_x
                ymin *= scale_y
                xmax *= scale_x
                ymax *= scale_y
                new_anno.append([bbox[0], xmin, ymin, xmax, ymax])

        elif i == 1:  # top-right
            img = cv2.resize(img, (output_size[1] - divid_point_x, divid_point_y))
            output_img[:divid_point_y, divid_point_x:output_size[1], :] = img
            for bbox in img_annos:
                xmin = bbox[1] - bbox[3]*0.5
                ymin = bbox[2] - bbox[4]*0.5
                xmax = bbox[1] + bbox[3]*0.5
                ymax = bbox[2] + bbox[4]*0.5

                xmin = scale_x + xmin * (1 - scale_x)
                ymin = ymin * scale_y
                xmax = scale_x + xmax * (1 - scale_x)
                ymax = ymax * scale_y
                new_anno.append([bbox[0], xmin, ymin, xmax, ymax])
        elif i == 2:  # bottom-left
            img = cv2.resize(img, (divid_point_x, output_size[0] - divid_point_y))
            output_img[divid_point_y:output_size[0], :divid_point_x, :] = img
            for bbox in img_annos:
                xmin = bbox[1] - bbox[3]*0.5
                ymin = bbox[2] - bbox[4]*0.5
                xmax = bbox[1] + bbox[3]*0.5
                ymax = bbox[2] + bbox[4]*0.5

                xmin = xmin * scale_x
                ymin = scale_y + ymin * (1 - scale_y)
                xmax = xmax * scale_x
                ymax = scale_y + ymax * (1 - scale_y)
                new_anno.append([bbox[0], xmin, ymin, xmax, ymax])
        else:  # bottom-right
            img = cv2.resize(img, (output_size[1] - divid_point_x, output_size[0] - divid_po
int_y))
            output_img[divid_point_y:output_size[0], divid_point_x:output_size[1], :] = im
g
            for bbox in img_annos:
                xmin = bbox[1] - bbox[3]*0.5
```

```
        ymin = bbox[2] - bbox[4]*0.5
        xmax = bbox[1] + bbox[3]*0.5
        ymax = bbox[2] + bbox[4]*0.5

        xmin = scale_x + xmin * (1 - scale_x)
        ymin = scale_y + ymin * (1 - scale_y)
        xmax = scale_x + xmax * (1 - scale_x)
        ymax = scale_y + ymax * (1 - scale_y)
        new_anno.append([bbox[0], xmin, ymin, xmax, ymax])

  if 0 < filter_scale:
    new_anno = [anno for anno in new_anno if
          filter_scale < (anno[3] - anno[1]) and filter_scale < (anno[4] - anno[2])]

  return output_img, new_anno


for i in range(723):
 img_paths, annos = get_dataset()
 idxs = random.sample(range(len(annos)), 4)
 new_image, new_annos = mosaic(img_paths, annos,
               idxs,
               OUTPUT_SIZE, SCALE_RANGE,
               filter_scale=FILTER_TINY_SCALE)

 cv2.imwrite('/content/gdrive/MyDrive/After Augmentation/Mosaic_pseudo/mosaic_be
fore' + str(i) + '.jpg', new_image) #The mosaic image
 # for anno in new_annos:
 #   start_point = (int(anno[1] * OUTPUT_SIZE[1]), int(anno[2] * OUTPUT_SIZE[0])
)
 #   end_point = (int(anno[3] * OUTPUT_SIZE[1]), int(anno[4] * OUTPUT_SIZE[0]))
 #   cv2.rectangle(new_image, start_point, end_point, (0, 255, 0), 1, cv2.LINE_AA)
 #   cv2.imwrite('output_box.jpg', new_image) # The mosaic image with the bounding
boxes

 yolo_anno = []

 for anno in new_annos:
  tmp = []
  tmp.append(anno[0])
  tmp.append((anno[3]+anno[1])/2)
  tmp.append((anno[4]+anno[2])/2)
  tmp.append(anno[3]-anno[1])
  tmp.append(anno[4]-anno[2])
  yolo_anno.append(tmp)
```

```
  with open('/content/gdrive/MyDrive/After Augmentation/Mosaic_pseudo/mosaic_befo
re'+ str(i) +  '.txt', 'w') as file: # The output annotation file will appear in the output.txt fi
le
    for line in yolo_anno:
      file.write((' ').join([str(x) for x in line]) + '\n')
```

In [ ]:

```
# writing image path in files
import glob
import os
from sklearn.model_selection import train_test_split

total_img = []

total_img.extend(glob.glob('/content/gdrive/MyDrive/Before Augmentation/TrainingDa
taset' + '/*.jpg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/After Augmentation/Brighting' + '
/*.jpg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/After Augmentation/Mosaic' + '/*
.jpg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/After Augmentation/CLAHE' + '/
*.jpg'))

total_img.extend(glob.glob('/content/gdrive/MyDrive/Before Augmentation/TestingDat
aset(pseudo)' + '/*.jpg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/Before Augmentation/TestingDat
aset(pseudo)' + '/*.jpeg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/After Augmentation/Brighting_ps
eudo' + '/*.jpg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/After Augmentation/Brighting_ps
eudo' + '/*.jpeg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/After Augmentation/Mosaic_pseu
do' + '/*.jpg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/After Augmentation/Mosaic_pseu
do' + '/*.jpeg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/After Augmentation/CLAHE_pse
udo' + '/*.jpg'))
total_img.extend(glob.glob('/content/gdrive/MyDrive/After Augmentation/CLAHE_pse
udo' + '/*.jpeg'))


print(len(total_img))

train, test =  train_test_split(total_img, shuffle=False)

print(len(train))
```

```
print(len(test))


os.chdir("..")
with open("/content/gdrive/MyDrive/After Augmentation/train.txt", "w") as outfile:
    for x in train:
        outfile.write(x)
        outfile.write("\n")
    outfile.close()
os.chdir("..")

os.chdir("..")
with open("/content/gdrive/MyDrive/After Augmentation/test.txt", "w") as outfile:
    for x in test:
        outfile.write(x)
        outfile.write("\n")
    outfile.close()
os.chdir("..")
```

In [ ]:

```
os.chdir("..")
with open("/content/gdrive/MyDrive/After Augmentation/rotation_total.txt", "w") as ou
tfile:
  # data = ['0','1','2','3','4']
  # count = [0,0,0,0,0]
  # temp = []
  for i in words:
    with open(i, "r") as f:
      content = f.readlines()
      for x in content:
        outfile.write(x)
  outfile.close()
os.chdir("..")
```

CLAHE

In [ ]:

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import matplotlib.image as mpimg

for img in words:
        prefix = img.split("/")[6].split(".")[0]
        ext = img.split("/")[6].split(".")[1]
```

```
        image = cv2.imread(img)
        image = cv2.resize(image, (500, 600))
        image_bw = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        clahe = cv2.createCLAHE(clipLimit = 5)
        final_img = clahe.apply(image_bw) + 30
        path = "/content/gdrive/MyDrive/After Augmentation/CLAHE_pseudo/C" + prefix
+ "." + ext
        mpimg.imsave(path, final_img)
```

In [ ]:

```
import shutil

for txt in words:
    prefix = txt.split("/")[6].split(".")[0]
    src = "/content/gdrive/MyDrive/Before Augmentation/TestingDataset(pseudo)/" + pre
fix + ".txt"
    for i in range(1):
        dst = "/content/gdrive/MyDrive/After Augmentation/CLAHE_pseudo/C" + prefix
+ ".txt"
        shutil.copyfile(src, dst)
```

YOLO installation

In [4]:

```
%cd /content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
```

```
/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet
```

In [5]:

```
!/usr/local/cuda/bin/nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Mon_Oct_12_20:09:46_PDT_2020
Cuda compilation tools, release 11.1, V11.1.105
Build cuda_11.1.TC455_06.29190527_0
```

In [6]:

```
!make
```

```
chmod +x *.sh
g++ -std=c++11 -std=c++11 -Iinclude/ -I3rdparty/stb/include -DOPENCV `pkg-config --cflags opencv4 2> /dev/null || pkg-config --cflags opencv` -DGPU -I/usr/local/cuda/include/ -DCUDNN
./src/image_opencv.cpp: In function ‘void draw_detections_cv_v3(void**, detection*, int, float, char**, image**, int, int)’:
./src/image_opencv.cpp:946:23: warning: variable ‘rgb’ set but not used [-Wunused-but-set-variable]
                 float rgb[3];
                       ^~~
./src/image_opencv.cpp: In function ‘void draw_train_loss(char*, void**, int, float, float, int, int, float, int, char*, float, int, int, double)’:
./src/image_opencv.cpp:1147:13: warning: this ‘if’ clause does not guard... [-Wmisleading-indentation]
             if (iteration_old == 0)
             ^~
./src/image_opencv.cpp:1150:10: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the ‘if’
          if (iteration_old != 0){
          ^~
./src/image_opencv.cpp: In function ‘void cv_draw_object(image, float*, int, int, int*, float*, int*, int, char**)’:
./src/image_opencv.cpp:1444:14: warning: unused variable ‘buff’ [-Wunused-variable]
         char buff[100];
              ^~~~
./src/image_opencv.cpp:1420:9: warning: unused variable ‘it_tb_res’ [-Wunused-variable]
     int it_tb_res = cv::createTrackbar(it_trackbar_name, window_name, &it_trackbar_value, 1000);
         ^~~~~~~~~
./src/image_opencv.cpp:1424:9: warning: unused variable ‘lr_tb_res’ [-Wunused-variable]
     int lr_tb_res = cv::createTrackbar(lr_trackbar_name, window_name, &lr_trackbar_value, 20);
         ^~~~~~~~~
./src/image_opencv.cpp:1428:9: warning: unused variable ‘cl_tb_res’ [-Wunused-variable]
     int cl_tb_res = cv::createTrackbar(cl_trackbar_name, window_name, &cl_trackbar_value, classes-1);
         ^~~~~~~~~
./src/image_opencv.cpp:1431:9: warning: unused variable ‘bo_tb_res’ [-Wunused-variable]
     int bo_tb_res = cv::createTrackbar(bo_trackbar_name, window_name, boxonly, 1);
```

Training

In [ ]:
YOLOv3

data = '/content/gdrive/MyDrive/AfterAugmentation/training_data/obj.data'
cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/yolo
v3.cfg'
output = '/content/gdrive/MyDrive/AfterAugmentation/training_data/yolov3/training_re
sult.txt'

!./darknet detector train $data $cfg yolov4.conv.137 -dont_show -ext_output > $output

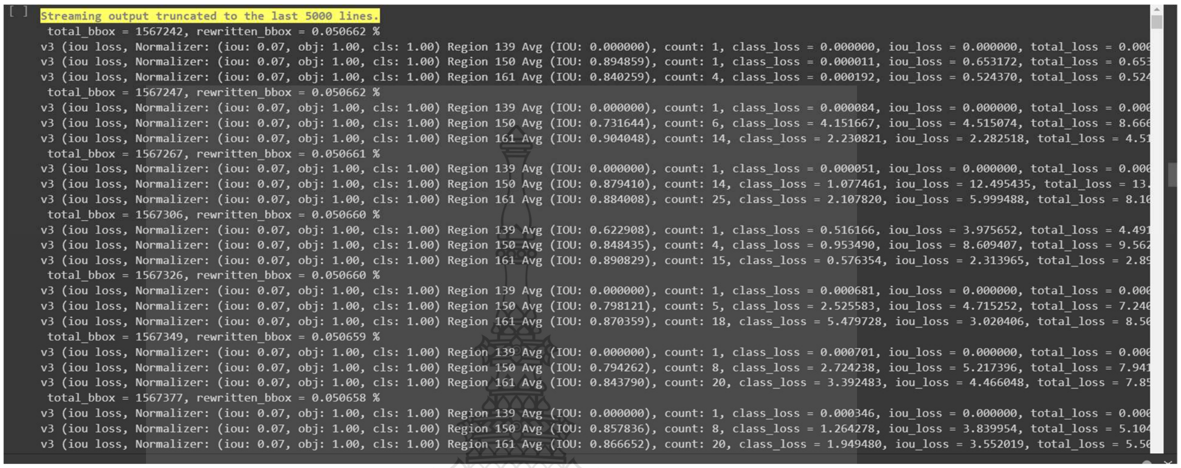
```
  total_bbox = 469828, rewritten_bbox = 0.029372 %
[ ] v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.790503), count: 9, class_loss = 2.468202, iou_loss = 0.743016, total_loss = 3.2112
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.008367, iou_loss = 0.000000, total_loss = 0.0083
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000006, iou_loss = 0.000000, total_loss = 0.000
     total_bbox = 469837, rewritten_bbox = 0.029372 %
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.831360), count: 3, class_loss = 1.352025, iou_loss = 0.408689, total_loss = 1.7607
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000009, iou_loss = 0.000000, total_loss = 0.0000
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000
     total_bbox = 469840, rewritten_bbox = 0.029372 %
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.751829), count: 2, class_loss = 0.818815, iou_loss = 0.395202, total_loss = 1.2140
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000004, iou_loss = 0.000000, total_loss = 0.0000
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000002, iou_loss = 0.000000, total_loss = 0.000
     total_bbox = 469842, rewritten_bbox = 0.029372 %
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.852301), count: 2, class_loss = 0.149767, iou_loss = 0.146614, total_loss = 0.2963
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000021, iou_loss = 0.000000, total_loss = 0.0000
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000001, iou_loss = 0.000000, total_loss = 0.000
     total_bbox = 469844, rewritten_bbox = 0.029371 %
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.825637), count: 2, class_loss = 0.382664, iou_loss = 0.120394, total_loss = 0.5030
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000012, iou_loss = 0.000000, total_loss = 0.0000
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000
     total_bbox = 469846, rewritten_bbox = 0.029371 %
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.905668), count: 3, class_loss = 0.672553, iou_loss = 0.061864, total_loss = 0.7344
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000452, iou_loss = 0.000000, total_loss = 0.0004
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000001, iou_loss = 0.000000, total_loss = 0.000
     total_bbox = 469849, rewritten_bbox = 0.029371 %
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.897162), count: 2, class_loss = 0.002187, iou_loss = 0.089455, total_loss = 0.0916
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000007, iou_loss = 0.000000, total_loss = 0.0000
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000
     total_bbox = 469851, rewritten_bbox = 0.029371 %
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.891309), count: 4, class_loss = 2.938421, iou_loss = 0.127958, total_loss = 3.0663
    v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000073, iou_loss = 0.000000, total_loss = 0.0000
```

In [ ]:
YOLOv4

data = '/content/gdrive/MyDrive/AfterAugmentation/training_data/obj.data'
cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/yolo
v4.cfg'
output = '/content/gdrive/MyDrive/AfterAugmentation/training_data/yolov4/training_re

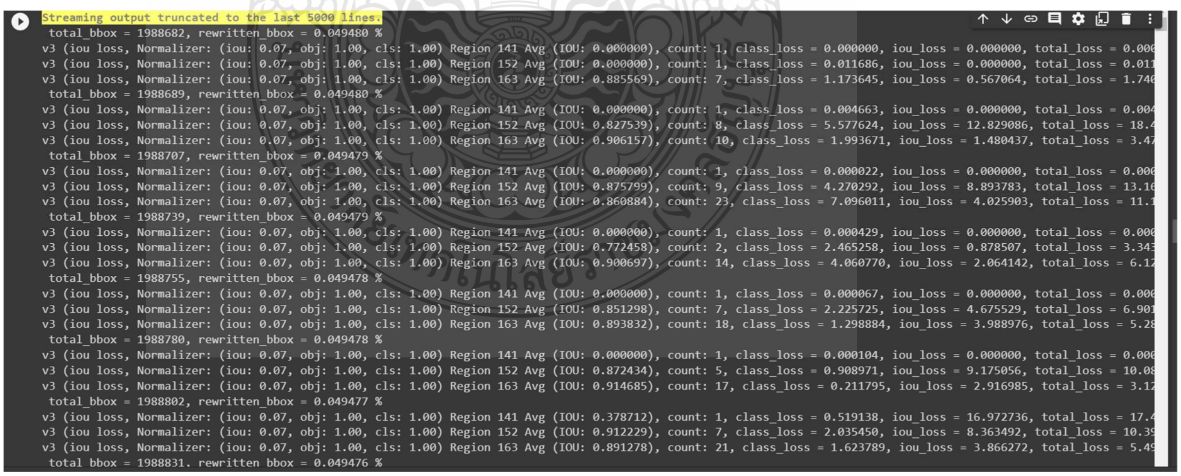
sult.txt'
weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_improved/yolov4_last.weights'

!./darknet detector train $data $cfg $weight -dont_show -ext_output > $output



In [ ]:
data = '/content/gdrive/MyDrive/AfterAugmentation/training_data/obj.data'
cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/fe1-yolov4.cfg'
output = '/content/gdrive/MyDrive/AfterAugmentation/training_data/feyolov4/training_result.txt'

!./darknet detector train $data $cfg yolov4.conv.137 -dont_show -ext_output > $output



Testing

In [8]:

```
file_names = get_file_names('/content/gdrive/MyDrive/AfterAugmentation/training_dat
a/EvaluationDataset')
print(file_names)
print(len(file_names))
```

```
['1.jpg', '2.jpg', '3.jpg', '4.jpg', '5.jpg', '6.jpg', '7.jpg', '8.jpg', '9.jpg', '10.jpg', '11.jpg', '12.png', '13.jpg', '14.jpg', '15.jpg', '16.jpg', '17.jpg', '18.
36
```

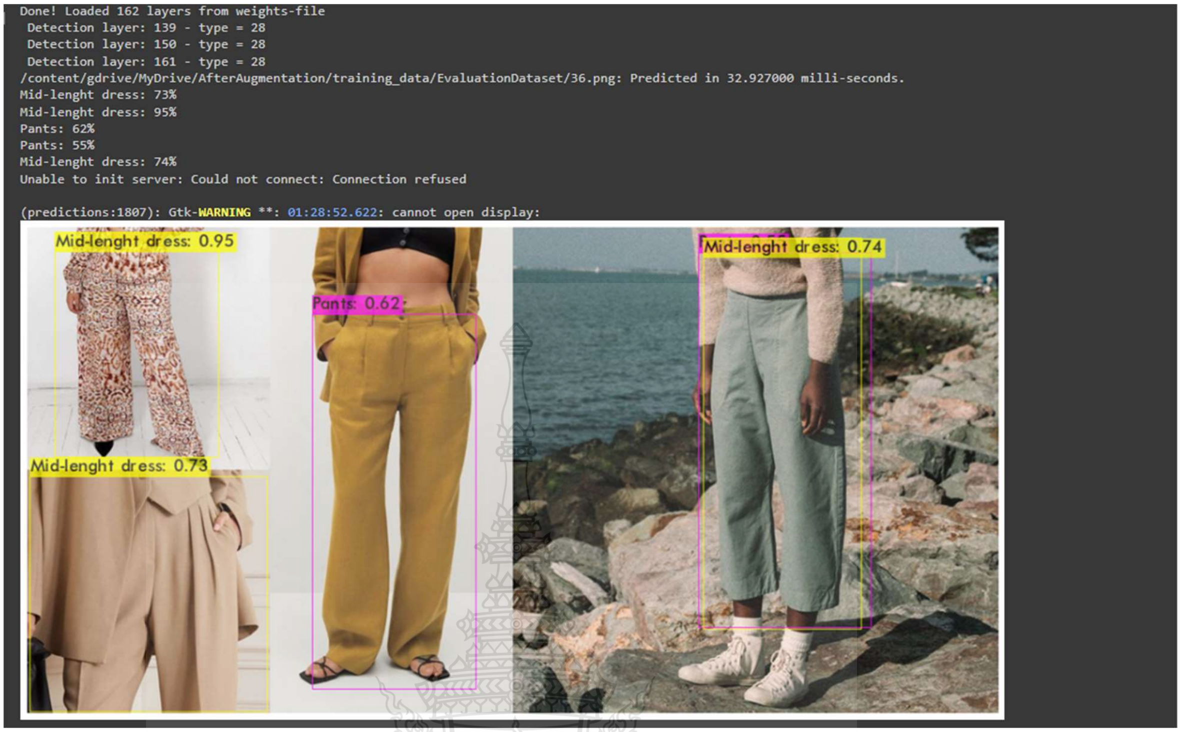
In [10]:

```
data = '/content/gdrive/MyDrive/AfterAugmentation/training_data/obj.data'

fe1cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/f
e1-yolov4.cfg'
yolov4cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/c
fg/yolov4.cfg'
yolov3cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/c
fg/yolov3.cfg'
yolov4resnext = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/dark
net/cfg/yolov4-resnext50-custom.cfg'

fe1weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_i
mproved/fe1-yolov4_last.weights'
yolov4weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backu
p_improved/yolov4_last.weights'
yolov3weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backu
p_improved/yolov3_last.weights'
yolov4resnext50 = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/Ex
celDataset/initback/yolov4-resnext50-custom_last.weights'


image = '/content/gdrive/MyDrive/AfterAugmentation/training_data/EvaluationDataset/'
+ file_names[35]
!./darknet detector test $data $yolov4cfg $yolov4weight $image -i 0 -thresh 0.5
imShow('predictions.jpg')
# download('predictions.jpg')
```



Model Evaluation

In [ ]:

```
data = '/content/gdrive/MyDrive/AfterAugmentation/training_data/obj.data'

fe1cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/fe1-yolov4.cfg'
yolov4cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/yolov4.cfg'
yolov3cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/yolov3.cfg'

fe1weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_improved/fe1-yolov4_last.weights'
yolov4weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_improved/yolov4_last.weights'
yolov3weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_improved/yolov3_last.weights'

!./darknet detector recall $data $yolov3cfg $yolov3weight -i 0 -thresh 0.5 -map
```

```
 106 yolo
[yolo] params: iou loss: mse (2), iou_norm: 0.75, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.00
Total BFLOPS 65.333
avg_outputs = 517519
 Allocate additional workspace_size = 52.43 MB
Loading weights from /content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_improved/yolov3_last.weights...
 seen 64, trained: 480 K-images (7 Kilo-batches_64)
Done! Loaded 107 layers from weights-file
    0     6     6       RPs/Img: 17.00  IOU: 78.00%     Recall:100.00%
    1    12    12       RPs/Img: 20.50  IOU: 76.15%     Recall:100.00%
    2    14    14       RPs/Img: 14.67  IOU: 77.24%     Recall:100.00%
    3    16    16       RPs/Img: 11.75  IOU: 78.07%     Recall:100.00%
    4    18    18       RPs/Img: 10.00  IOU: 78.54%     Recall:100.00%
    5    21    21       RPs/Img: 9.33   IOU: 80.25%     Recall:100.00%
    6    24    24       RPs/Img: 9.29   IOU: 81.44%     Recall:100.00%
    7    27    27       RPs/Img: 9.12   IOU: 82.39%     Recall:100.00%
    8    28    28       RPs/Img: 9.22   IOU: 82.43%     Recall:100.00%
    9    29    29       RPs/Img: 9.20   IOU: 82.46%     Recall:100.00%
   10    30    30       RPs/Img: 8.91   IOU: 82.69%     Recall:100.00%
   11    32    32       RPs/Img: 8.67   IOU: 83.28%     Recall:100.00%
   12    34    34       RPs/Img: 8.31   IOU: 83.66%     Recall:100.00%
   13    36    36       RPs/Img: 8.14   IOU: 84.14%     Recall:100.00%
   14    39    39       RPs/Img: 7.80   IOU: 84.62%     Recall:100.00%
   15    42    42       RPs/Img: 7.50   IOU: 85.06%     Recall:100.00%
   16    45    45       RPs/Img: 7.24   IOU: 85.43%     Recall:100.00%
   17    46    46       RPs/Img: 7.72   IOU: 85.45%     Recall:100.00%
   18    47    47       RPs/Img: 8.05   IOU: 85.47%     Recall:100.00%
   19    48    48       RPs/Img: 8.50   IOU: 85.51%     Recall:100.00%
   20    49    49       RPs/Img: 8.48   IOU: 85.44%     Recall:100.00%
   21    50    50       RPs/Img: 8.41   IOU: 84.93%     Recall:100.00%
   22    51    51       RPs/Img: 8.35   IOU: 84.44%     Recall:100.00%
   23    54    54       RPs/Img: 8.42   IOU: 83.71%     Recall:100.00%
   24    57    57       RPs/Img: 8.44   IOU: 82.99%     Recall:100.00%
   25    60    60       RPs/Img: 8.42   IOU: 82.33%     Recall:100.00%
   26    62    62       RPs/Img: 8.26   IOU: 82.03%     Recall:100.00%
   27    64    64       RPs/Img: 8.11   IOU: 81.73%     Recall:100.00%
   28    66    66       RPs/Img: 7.97   IOU: 81.47%     Recall:100.00%
   29    68    68       RPs/Img: 7.80   IOU: 81.49%     Recall:100.00%
   30    70    70       RPs/Img: 7.65   IOU: 81.50%     Recall:100.00%
   31    72    72       RPs/Img: 7.50   IOU: 81.51%     Recall:100.00%
   32    74    74       RPs/Img: 7.55   IOU: 81.51%     Recall:100.00%
   33    76    76       RPs/Img: 7.56   IOU: 81.48%     Recall:100.00%
```

```
In [ ]:
data = '/content/gdrive/MyDrive/AfterAugmentation/training_data/obj.data'

fe1cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/fe1-yolov4.cfg'
yolov4cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/yolov4.cfg'
yolov3cfg = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/darknet/cfg/yolov3.cfg'

fe1weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_improved/fe1-yolov4_last.weights'
yolov4weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_improved/yolov4_last.weights'
yolov3weight = '/content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_improved/yolov3_last.weights'

!./darknet detector map $data $yolov4cfg $yolov4weight -i 0 -thresh 1
```

```
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.10
nms_kind: greedynms (1), beta = 0.600000
 151 route  147                                        ->   26 x  26 x 256
 152 conv    512       3 x 3/ 2     26 x  26 x 256 ->   13 x  13 x 512 0.399 BF
 153 route  152 116                                ->   13 x  13 x1024
 154 conv    512       1 x 1/ 1     13 x  13 x1024 ->   13 x  13 x 512 0.177 BF
 155 conv   1024       3 x 3/ 1     13 x  13 x 512 ->   13 x  13 x1024 1.595 BF
 156 conv    512       1 x 1/ 1     13 x  13 x1024 ->   13 x  13 x 512 0.177 BF
 157 conv   1024       3 x 3/ 1     13 x  13 x 512 ->   13 x  13 x1024 1.595 BF
 158 conv    512       1 x 1/ 1     13 x  13 x1024 ->   13 x  13 x 512 0.177 BF
 159 conv   1024       3 x 3/ 1     13 x  13 x 512 ->   13 x  13 x1024 1.595 BF
 160 conv     30       1 x 1/ 1     13 x  13 x1024 ->   13 x  13 x  30 0.010 BF
 161 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
Total BFLOPS 59.592
avg_outputs = 490304
 Allocate additional workspace_size = 52.43 MB
Loading weights from /content/gdrive/MyDrive/MasterThesis/ImageProcessing/Shopee/backup_improved/yolov4_last.weights...
 seen 64, trained: 480 K-images (7 Kilo-batches_64)
Done! Loaded 162 layers from weights-file

 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
4692
 detections_count = 14643, unique_truth_count = 8280
class_id = 0, name = Pants, ap = 93.95%          (TP = 0, FP = 0)
class_id = 1, name = Mid-lenght dress, ap = 96.84%       (TP = 0, FP = 0)
class_id = 2, name = Hoodie, ap = 95.24%         (TP = 0, FP = 0)
class_id = 3, name = Jacket, ap = 97.76%         (TP = 0, FP = 0)
class_id = 4, name = Mid-length skirt, ap = 95.58%       (TP = 0, FP = 0)

 for conf_thresh = 1.00, precision = -nan, recall = 0.00, F1-score = -nan
 for conf_thresh = 1.00, TP = 0, FP = 0, FN = 8280, average IoU = 0.00 %
```

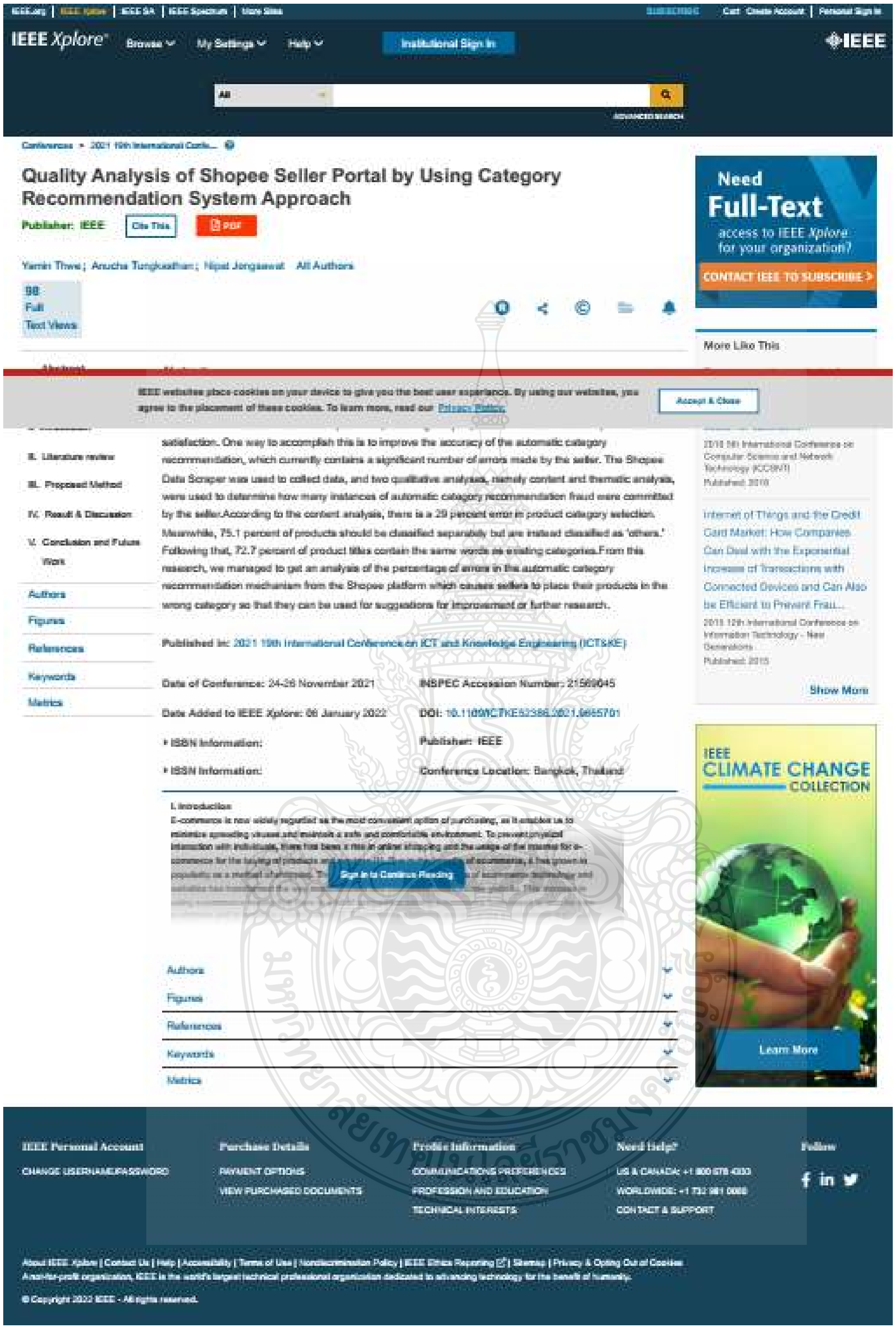
APPENDIX B.

INTERNATIONAL PUBLICATION

Thwe, Y., Jongsawat, N. and Tungkasthan, A., 2022. A Semi-Supervised Learning Approach for Automatic Detection and Fashion Product Category Prediction with Small Training Dataset Using FC-YOLOv4. Applied Sciences, 12(16), p.8068. https://doi: 10.1109/ICTKE52386.2021.9665701 (Scopus)

Thwe, Y.; Jongsawat, N.; Tungkasthan, A., A Semi-Supervised Learning Approach for Automatic Detection and Fashion Product Category Prediction with Small Training Dataset Using FC-YOLOv4. Appl. Sci. 2022, 12, 8068. https://doi.org/10.3390/app12168068 (Scopus: Q2, Web of Science: Q2)

Thwe, Y.; Jongsawat, N.; Tungkasthan, A., AFAD: Accurate Fashion and Accessories Detection for Mobile Application Based on Deep Learning. International Journal of Electrical and Computer Engineering (IJECE).Vol. 13, No. 4, August2023,pp.4347~4356,\ISSN:2088-8708, https://doi.org/10.11591/ijece.v13i4.pp4347-4356(Scopus: Q2)



IEEE Xplore

Conferences > 2021 19th International Confe...

Quality Analysis of Shopee Seller Portal by Using Category Recommendation System Approach

Publisher: IEEE | Cite This | PDF

Yamin Thwe ; Anucha Tungkasthan ; Nipat Jongsawat All Authors

98 Full Text Views

satisfaction. One way to accomplish this is to improve the accuracy of the automatic category recommendation, which currently contains a significant number of errors made by the seller. The Shopee Data Scraper was used to collect data, and two qualitative analyses, namely content and thematic analysis, were used to determine how many instances of automatic category recommendation fraud were committed by the seller. According to the content analysis, there is a 29 percent error in product category selection. Meanwhile, 75.1 percent of products should be classified separately but are instead classified as 'others.' Following that, 72.7 percent of product titles contain the same words as existing categories. From this research, we managed to get an analysis of the percentage of errors in the automatic category recommendation mechanism from the Shopee platform which causes sellers to place their products in the wrong category so that they can be used for suggestions for improvement or further research.

Published in: 2021 19th International Conference on ICT and Knowledge Engineering (ICT&KE)

Date of Conference: 24-26 November 2021 | INSPEC Accession Number: 21569045

Date Added to IEEE Xplore: 06 January 2022 | DOI: 10.1109/ICTKE52386.2021.9665701

ISBN Information: | Publisher: IEEE

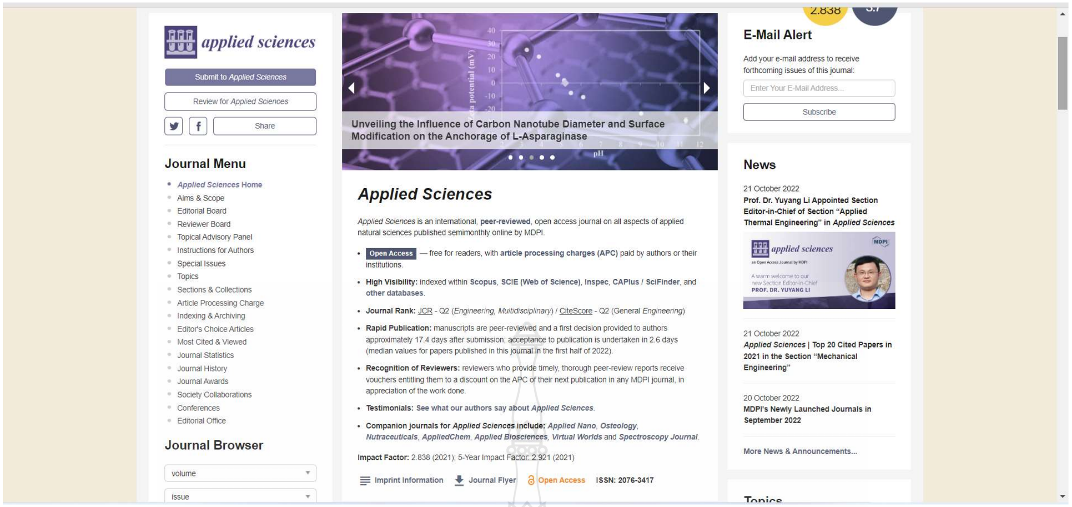
ISSN Information: | Conference Location: Bangkok, Thailand

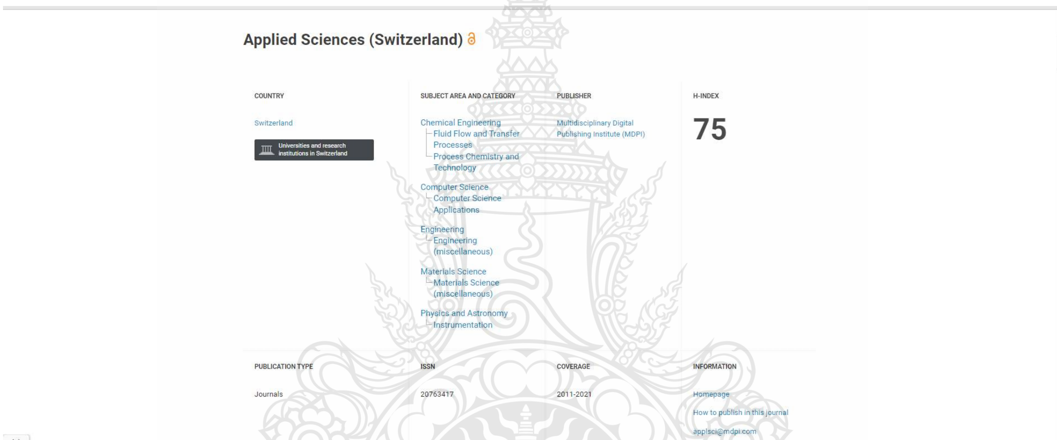
I. Introduction

E-commerce is now widely regarded as the most convenient option of purchasing, as it enables us to minimize spreading viruses and maintain a safe and comfortable environment. To prevent physical interaction with individuals, there has been a rise in online shopping and the usage of the internet for e-commerce for the buying of products ...

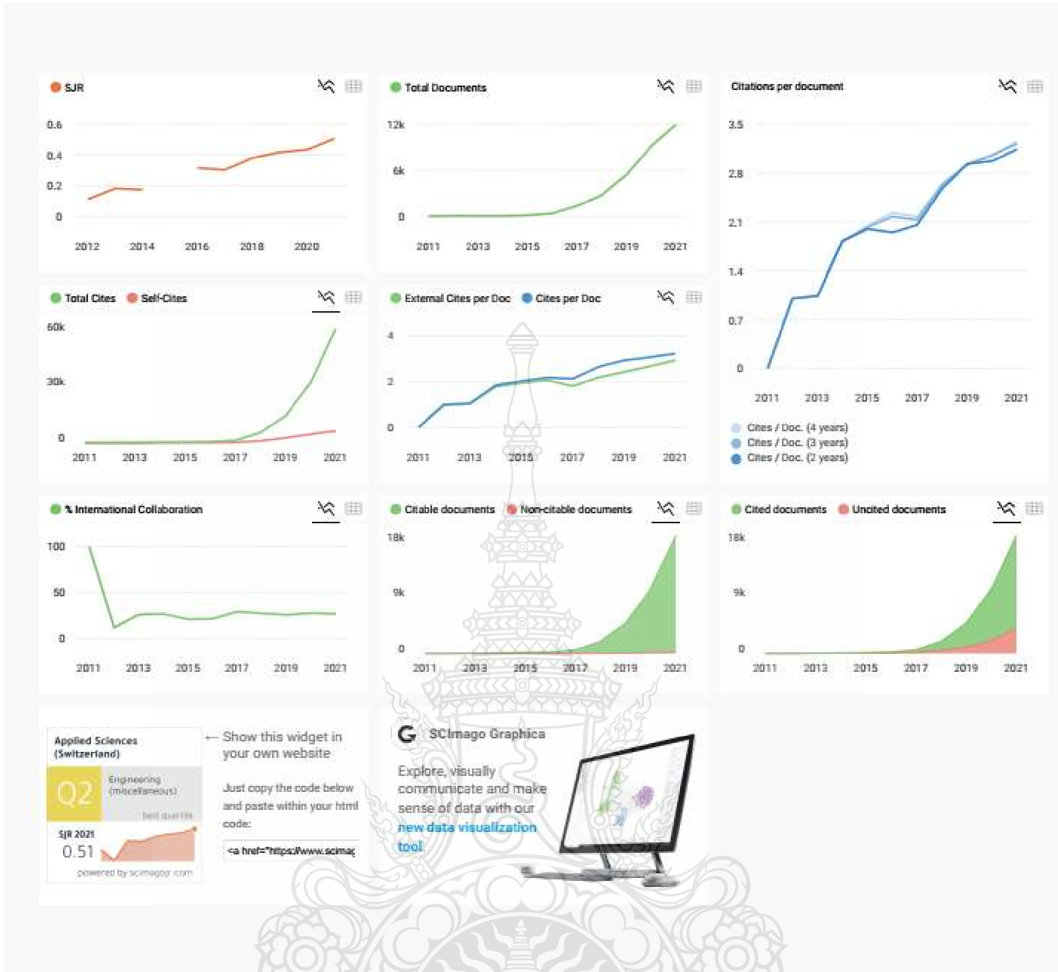
Sign in to Continue Reading

Authors | Figures | References | Keywords | Metrics


applied sciences
Submit to Applied Sciences
Review for Applied Sciences
Share
Journal Menu
Applied Sciences Home
Aims & Scope
Editorial Board
Reviewer Board
Topical Advisory Panel
Instructions for Authors
Special Issues
Topics
Sections & Collections
Article Processing Charge
Indexing & Archiving
Editor's Choice Articles
Most Cited & Viewed
Journal Statistics
Journal History
Journal Awards
Society Collaborations
Conferences
Editorial Office
Journal Browser
volume
issue
Unveiling the Influence of Carbon Nanotube Diameter and Surface Modification on the Anchorage of L-Asparaginase
Applied Sciences
Applied Sciences is an international, peer-reviewed, open access journal on all aspects of applied natural sciences published semimonthly online by MDPI.
Open Access — free for readers, with article processing charges (APC) paid by authors or their institutions.
High Visibility: indexed within Scopus, SCIE (Web of Science), Inspec, CAPlus / SciFinder, and other databases.
Journal Rank: JCR - Q2 (Engineering, Multidisciplinary) / CiteScore - Q2 (General Engineering)
Rapid Publication: manuscripts are peer-reviewed and a first decision provided to authors approximately 17.4 days after submission; acceptance to publication is undertaken in 2.6 days (median values for papers published in this journal in the first half of 2022).
Recognition of Reviewers: reviewers who provide timely, thorough peer-review reports receive vouchers entitling them to a discount on the APC of their next publication in any MDPI journal, in appreciation of the work done.
Testimonials: See what our authors say about Applied Sciences.
Companion journals for Applied Sciences include: Applied Nano, Osteology, Nutraceuticals, AppliedChem, Applied Biosciences, Virtual Worlds and Spectroscopy Journal.
Impact Factor: 2.838 (2021); 5-Year Impact Factor: 2.921 (2021)
Imprint Information
Journal Flyer
Open Access
ISSN: 2076-3417
2.838
E-Mail Alert
Add your e-mail address to receive forthcoming issues of this journal:
Enter Your E-Mail Address...
Subscribe
News
21 October 2022
Prof. Dr. Yuyang Li Appointed Section Editor-in-Chief of Section "Applied Thermal Engineering" in Applied Sciences
21 October 2022
Applied Sciences | Top 20 Cited Papers in 2021 in the Section "Mechanical Engineering"
20 October 2022
MDPI's Newly Launched Journals in September 2022
More News & Announcements...


Applied Sciences (Switzerland)
COUNTRY
Switzerland
Universities and research institutions in Switzerland
SUBJECT AREA AND CATEGORY
Chemical Engineering
Fluid Flow and Transfer Processes
Process Chemistry and Technology
Computer Science
Computer Science Applications
Engineering
Engineering (miscellaneous)
Materials Science
Materials Science (miscellaneous)
Physics and Astronomy
Instrumentation
PUBLISHER
Multidisciplinary Digital Publishing Institute (MDPI)
H-INDEX
75
PUBLICATION TYPE
Journals
ISSN
20763417
COVERAGE
2011-2021
INFORMATION
Homepage
How to publish in this journal
applsci@mdpi.com


Quartiles
Computer Science Applications
Engineering (miscellaneous)
Fluid Flow and Transfer Processes
Instrumentation
Materials Science (miscellaneous)
Process Chemistry and Technology
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021


SJR
Total Documents
Citations per document
Total Cites
Self-Cites
External Cites per Doc
Cites per Doc
Cites / Doc. (4 years)
Cites / Doc. (3 years)
Cites / Doc. (2 years)
% International Collaboration
Citable documents
Non-citable documents
Cited documents
Uncited documents
Applied Sciences (Switzerland)
Q2
Engineering (miscellaneous)
best quartile
SJR 2021
0.51
powered by scimagojr.com
← Show this widget in your own website
Just copy the code below and paste within your html code:
<a href="https://www.scimag
SCImago Graphica
Explore, visually communicate and make sense of data with our new data visualization tool.

Scopus Preview   Author Search   Sources   Create account   Sign in

Source details

Feedback >   Compare sources >

Applied Sciences (Switzerland)

Open Access

Scopus coverage years: from 2011 to Present

Publisher: Multidisciplinary Digital Publishing Institute (MDPI)

E-ISSN: 2076-3417

Subject area: Engineering: General Engineering | Physics and Astronomy: Instrumentation | Computer Science: Computer Science Applications | Chemical Engineering: Fluid Flow and Transfer Processes | Materials Science: General Materials Science | View all

Source type: Journal

View all documents >   Set document alert   Save to source list   Source Homepage

CiteScore 2021
3.7

SJR 2021
0.507

SNIP 2021
1.026

CiteScore   CiteScore rank & trend   Scopus content coverage

Improved CiteScore methodology

CiteScore 2021 counts the citations received in 2018-2021 to articles, reviews, conference papers, book chapters and data papers published in 2018-2021, and divides this by the number of publications published in 2018-2021. Learn more >

CiteScore 2021

3.7 = 107,111 Citations 2018 - 2021 / 28,948 Documents 2018 - 2021

Calculated on 05 May, 2022

CiteScoreTracker 2022

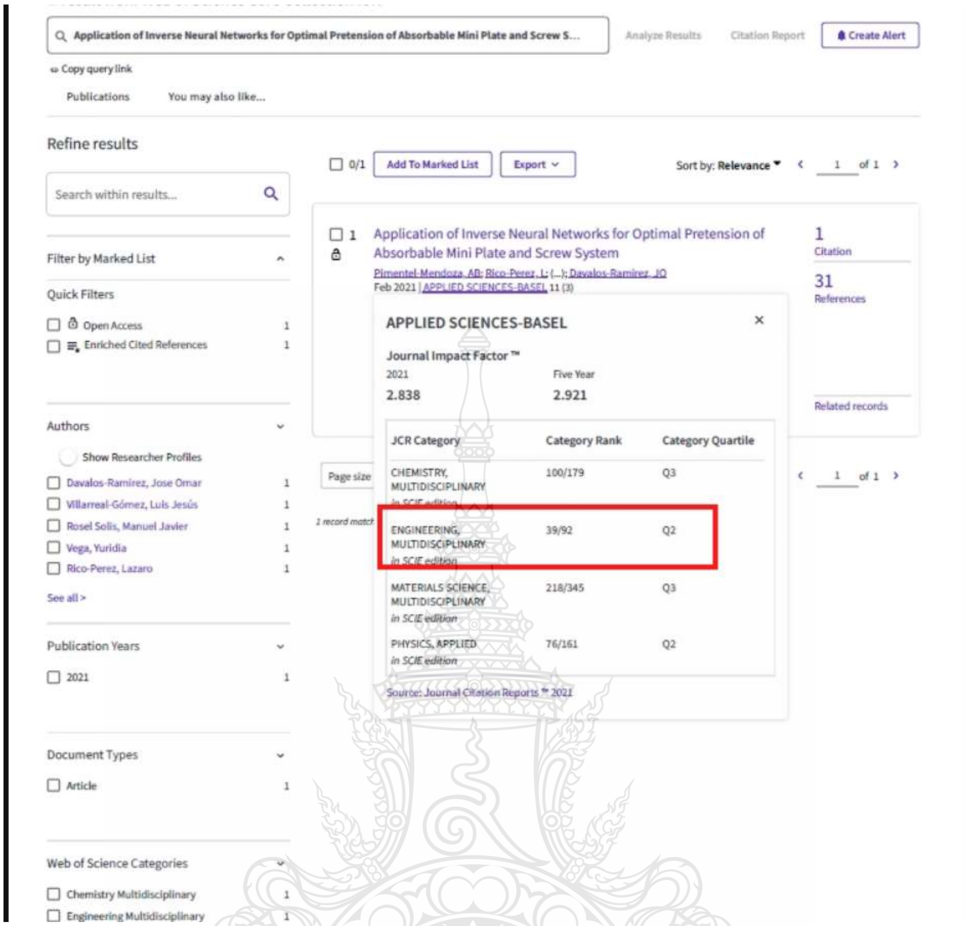
4.0 = 142,262 Citations to date / 35,369 Documents to date

Last updated on 05 October, 2022 • Updated monthly

CiteScore rank 2021

Category	Rank	Percentile
Engineering — General Engineering	#81/300	73rd
Physics and Astronomy — Instrumentation	#52/136	62nd
Computer Science — Computer Science Applications	#301/747	59th

View CiteScore methodology >   CiteScore FAQ >   Add CiteScore to your site


Application of Inverse Neural Networks for Optimal Pretension of Absorbable Mini Plate and Screw S...
Analyze Results
Citation Report
Create Alert
Copy query link
Publications
You may also like...
Refine results
Search within results...
Filter by Marked List
Quick Filters
Open Access 1
Enriched Cited References 1
Authors
Show Researcher Profiles
Davalos-Ramirez, Jose Omar 1
Villarreal-Gómez, Luis Jesús 1
Rosel Solis, Manuel Javier 1
Vega, Yuridia 1
Rico-Perez, Lazaro 1
See all >
Publication Years
2021 1
Document Types
Article 1
Web of Science Categories
Chemistry Multidisciplinary 1
Engineering Multidisciplinary 1
0/1
Add To Marked List
Export
Sort by: Relevance
1 of 1
1
Application of Inverse Neural Networks for Optimal Pretension of Absorbable Mini Plate and Screw System
Pimentel-Mendoza, AB; Rico-Perez, L; (...); Davalos-Ramirez, JO
Feb 2021 | APPLIED SCIENCES-BASEL 11 (3)
1
Citation
31
References
Related records
Page size
1 record match
APPLIED SCIENCES-BASEL
Journal Impact Factor ™
2021
2.838
Five Year
2.921
JCR Category
Category Rank
Category Quartile
CHEMISTRY, MULTIDISCIPLINARY in SCIE edition 100/179 Q3
ENGINEERING, MULTIDISCIPLINARY in SCIE edition 39/92 Q2
MATERIALS SCIENCE, MULTIDISCIPLINARY in SCIE edition 218/345 Q3
PHYSICS, APPLIED in SCIE edition 76/161 Q2
Source: Journal Citation Reports ™ 2021


MDPI
Search for Articles:
applied sciences
Journal Menu
Applied Sciences — Indexing & Archiving
Indexing & Abstracting Services
We use cookies on our website to ensure you get the best experience.
Read more about our cookies here.
Journal Browser
Journal Ranking Lists
Journal Directories
Repository delivery services
Digital Preservation
Content Aggregators

applied sciences

Submit to Special Issue

Submit Abstract to Special Issue

Review for *Applied Sciences*

Edit a Special Issue

Journal Menu

- *Applied Sciences* Home
- Aims & Scope
- Editorial Board

- Topics
- Sections & Collections
- Article Processing Charge
- Indexing & Archiving
- Editor's Choice Articles
- Most Cited & Viewed
- Journal Statistics
- Journal History
- Journal Awards
- Society Collaborations
- Conferences
- Editorial Office

Journal Browser

volume

issue

Go

> Forthcoming issue
> Current issue

Vol. 12 (2022) Vol. 11 (2021) Vol. 10 (2020) Vol. 9 (2019) Vol. 8 (2018) Vol. 7 (2017) Vol. 6 (2016) Vol. 5 (2015) Vol. 4 (2014) Vol. 3 (2013) Vol. 2 (2012) Vol. 1 (2011)

Special Issue "Applications of Deep Learning and Artificial Intelligence Methods"

- Print Special Issue Flyer
- Special Issue Editors
- Special Issue Information
- Keywords
- Published Papers
- Planned Papers

A special issue of *Applied Sciences* (ISSN 2076-3417). This special issue belongs to the section "Computing and Artificial

Share This Special Issue

Special Issue Editors

Prof. Dr. Yujin Lim E-Mail Website
Guest Editor
Department of IT Engineering, Sookmyung Women's University, Cheongpa-ro 47-gil 100, Yongsan-gu, Seoul 14310, Korea
Interests: multi-agent system; artificial intelligence
Special Issues, Collections and Topics in MDPI journals

Prof. Dr. Hideyuki Takahashi E-Mail Website
Guest Editor
Department of Information Science, Faculty of Liberal Arts, Tohoku Gakuin University, 2-1-1 Tenjinzawa, Izumi-ku, Sendai, Miyagi 981-3193, Japan
Interests: Internet of Things; ubiquitous computing; multi-agent system
Special Issues, Collections and Topics in MDPI journals

Special Issue Information

Dear Colleagues,

Deep Learning and Artificial Intelligence have attracted great attention in almost every field in recent years. Applications of deep learning and artificial intelligence methods is now pervasive into many various fields beyond the conventional computer engineering areas. Therefore, the goal of this special issue is intended for discussing new ideas and recent experimental results in the field of applications of deep learning and artificial intelligence methods.

Topics of interest include, but are not limited to, the following:

-Artificial Intelligence Tools & Applications

-Automatic Control

-Natural Language Processing

-Computer Vision and Speech Understanding

-Data Mining and Analysis

-Heuristic and AI Planning Strategies

-Intelligent System

-Robotics

Prof. Dr. Yujin Lim
Prof. Dr. Hideyuki Takahashi
Guest Editors

Open Access Article

A Semi-Supervised Learning Approach for Automatic Detection and Fashion Product Category Prediction with Small Training Dataset Using FC-YOLOv4

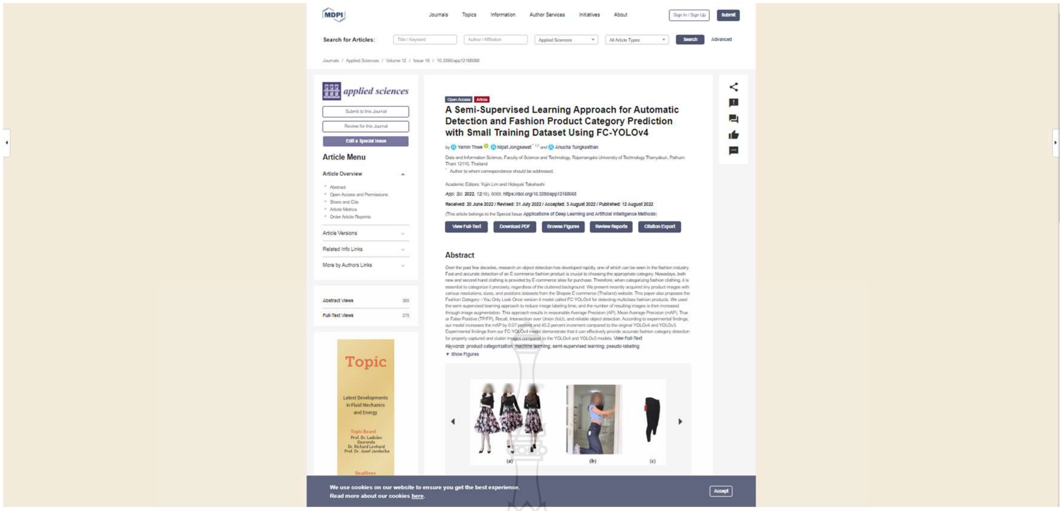
by Yamin Thwe, Nipat Jongsawat and Anucha Tungkasthan

Appl. Sci. 2022, *12*(16), 8068; https://doi.org/10.3390/app12168068 - 12 Aug 2022

Abstract Over the past few decades, research on object detection has developed rapidly, one of which can be seen in the fashion industry. Fast and accurate detection of an E-commerce fashion product is crucial to choosing the appropriate category. Nowadays, both new and second-hand [...] Read more.

(This article belongs to the Special Issue Applications of Deep Learning and Artificial Intelligence Methods)

► Show Figures


Open Access Article

A Semi-Supervised Learning Approach for Automatic Detection and Fashion Product Category Prediction with Small Training Dataset Using FC-YOLOv4

by Yamin Thwe, Nipat Jongsawat * and Anucha Tungkasthan

Data and Information Science, Faculty of Science and Technology, Rajamangala University of Technology Thanyaburi, Pathum Thani 12110, Thailand

* Author to whom correspondence should be addressed.

Academic Editors: Yujin Lim and Hideyuki Takahashi

Appl. Sci. 2022, 12(16), 8068; https://doi.org/10.3390/app12168068

Received: 20 June 2022 / Revised: 31 July 2022 / Accepted: 3 August 2022 / Published: 12 August 2022

(This article belongs to the Special Issue Applications of Deep Learning and Artificial Intelligence Methods)

Abstract

Over the past few decades, research on object detection has developed rapidly, one of which can be seen in the fashion industry. Fast and accurate detection of an E-commerce fashion product is crucial to choosing the appropriate category. Nowadays, both new and second-hand clothing is provided by E-commerce sites for purchase. Therefore, when categorizing fashion clothing, it is essential to categorize it precisely, regardless of the cluttered background. We present recently acquired tiny product images with various resolutions, sizes, and positions datasets from the Shopee E-commerce (Thailand) website. This paper also proposes the Fashion Category—You Only Look Once version 4 model called FC-YOLOv4 for detecting multiclass fashion products. We used the semi-supervised learning approach to reduce image labeling time, and the number of resulting images is then increased through image augmentation. This approach results in reasonable Average Precision (AP), Mean Average Precision (mAP), True or False Positive (TP/FP), Recall, Intersection over Union (IoU), and reliable object detection. According to experimental findings, our model increases the mAP by 0.07 percent and 40.2 percent increment compared to the original YOLOv4 and YOLOv3. Experimental findings from our FC-YOLOv4 model demonstrate that it can effectively provide accurate fashion category detection for properly captured and clutter images compared to the YOLOv4 and YOLOv3 models. View Full-Text

Keywords: product categorization; machine learning; semi-supervised learning; pseudo-labeling

11:10 AM Wed 21 Dec 76%

Inbox
yamin_t@mail.rmutt.ac.th

RE: [IJECE] Please Revise Your Paper

AS Azidanti Saufi 14 Dec
To You, ผู้ช่วยศาสตราจารย์ ดร.นิพัทธ์ จงสวัสดิ์ and อนุชา ตุงคัษฐาน

29017 8%
DOCX - 5.2 MB

Dear Authors,

Hello, how are you?
We wish you very well and we hope this email reaches you well.

We are glad to inform you that your paper is in the layout stage for possible publication in the forthcoming issue (Vol 13 No 4) of this journal. Your cooperation for final checking and/or updating your paper is required. Please find the attached file (including comments and/or marked parts) to take further actions. Please revise in the file only and don't make another file. Kindly submit your updated paper maximal Dec 20, 2022 by replying to this email.

Please note that this email is only assigned for layout and editing purposes. For other communication purposes, reach us through the principal contact of the journal.

Your cooperation is highly appreciated. Thank you

Kind Regards,
Azidanti Saufi
Editorial Staff on behalf of Editor-in-Chief
Layout and Editing Team
International Journal of Electrical and Computer Engineering (IJECE)
Institute of Advanced Engineering and Science (IAES)
ISSN: 2088-8708
http://ijece.iaescore.com/index.php/IJECE/index
email: ijece@iaesjournal.com, ijece@iaescore.com

You 15 Dec
To Azidanti Saufi

Reply to All

Mail Search 21 Calendar

Biography

Name – Surname	Yamin Thwe
Date of Birth	March 19, 1997
Address	39 Rangsit, Thanyaburi District, Chang Wat Pathum Thani 12110
Education	Bachelor of Technology in Information Technology – Technological University Hmawbi, from 2013 - 2019. GPA: 4.83.
	Bachelor of Engineering in Information Technology – Technological University Hmawbi, from 2019 - 2020. GPA: 4.52.
Experience Work	MSDT Software Company Limited – Senior Programmer (2019- 2021)
Telephone Number	+66641430243
Email Address	yamin_t@mail.rmutt.ac.th