成都理工大學中英合作办学
CDUT Sino-British Collaborative Education

OXFORD BROOKES UNIVERSITY

UNDERGRADUATE PROJECT REPORT

Project Title:	**Fashion Items Recognition using Deep Learning**
Surname:	**Liu**
First Name:	**Zhiyi**
Student Number:	**202018020319**
Supervisor Name:	**Dr.Joojo Walker**
Module Code:	**CHC 6096**
Module Name:	**Project**
Date Submitted:	**May 6, 2024**

Chengdu University of Technology Oxford Brookes College

Chengdu University of Technology

BSc (Single Honours) Degree Project

Programme Name: **Software Engineering**

Module No.: **CHC 6096**

Surname: ……Liu……

First Name: ……ZhiYi……

Project Title: ……Fashion item recognition using deep learning……

Student No.: ……202018020319……

Supervisor:……Dr.Joojo Walker……

2ND Supervisor (if applicable): **Not Applicable**

Date submitted: **May 6, 2024**

A report submitted as part of the requirements for the degree of BSc (Hons) in Software Engineering

At

Chengdu University of Technology Oxford Brookes College

Declaration

Student Conduct Regulations:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed online at
https://www.brookes.ac.uk/students/sirt/student-conduct/

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them.

Signature ……. Date ……**May 6, 2024**……

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library.

Signature ……. Date ……**May 6, 2024**……

Acknowledgment

I extend my deepest respect to all those who offered their support and assistance throughout my academic journey, contributing significantly to the successful completion of my final report. Their guidance and encouragement have been the base of my success.

My first appreciation goes to Oxford Brookes University for creating an high quality learning environment that has been instrumental in my professional and personal development. The rich experiences, valuable lessons, and the supportive study resource provided by the staff members have helped me a lot on my academic journey. Their constant guidance has been a beacon, illuminating my path to growth and learning.

A special acknowledgment is showed for my supervisor, Dr. Joojo Walker, whose teaching has been the most important part of strength. His patience, high standards, and strong support lead me forward in my project. His useful advice and constructive criticism on various aspects, from topic selection to research methodology and writing, have been invaluable. Dr. Walker's exemplary conduct and words of wisdom have imbued me with a spirit of diligence and a drive for continual learning that I will carry into my future academic pursuits and professional career.

I owe a heartfelt thanks to my parents, who have been my bedrock of support. In times of challenge, their sacrifices and hard work have been the guiding forces behind my accomplishments. Their enduring love and encouragement have been the source of my strength and motivation. I am eternally grateful for their selfless devotion and support.

Finally, I extend my thanks again to all who have supported and believed in me throughout this journey.

Table of Contents

Abstract

In recent years, the fashion industry has seen a significant transformation through the integration of advanced technologies.This project research delves into the burgeoning intersection of deep learning technologies and the fashion industry, particularly focusing on the utilization of advanced computer vision techniques for fashion item recognition. The study harnesses a rich dataset of 43,983 images spanning 87 distinct fashion classes to train and analyze seven prominent deep learning models: ResNet, AlexNet, ConvNeXt, DenseNet, EfficientNet, MobileNet, and Vision Transformer. Each model's performance was rigorously assessed based on accuracy, loss, precision, recall, and F1-score, enabling a detailed comparison to determine the most efficient models for practical application. A Flask-based web application was developed to demonstrate real-world applicability, allowing users to upload images for immediate fashion item identification. This research highlights the potential of convolutional neural networks (CNNs) and other deep learning frameworks in enhancing automation, thereby streamlining inventory management and improving consumer shopping experiences through more accurate search results and recommendations. Despite progress, the study identifies ongoing challenges, including handling the vast diversity within fashion items and selecting optimal models for maximum accuracy, particularly in diverse and dynamic environments like those encountered in global fashion companies such as Adidas AG™. The findings point towards continuous refinement of model architectures and training processes as critical to overcoming these challenges, setting a clear path for future advancements in the field.

Keywords: Deep Learning, Fashion Item Recognition, Flask-Web Application Development, ResNet, AlexNet, ConvNeXt, DenseNet, EfficientNet, MobileNet, Vision Transform

Abbreviations

Abbreviations	Definition
CNN	Convolutional Neural Network
Resnet	Residual Network
ConveNeXt	Convolutional Neural Networks Next
DenseNet	Densely Connected Convolutional Networks
ViT	Vision Transformer
LRN	Local Response Normalization
RLUE	Rectified Linear Unit
GELU	Gaussian Error Linear Unit
DNN	Deep Neural Networks
NLP	Natural Language Processing
MBConv	Mobile Inverted Bottleneck Convolution
FFNN	Feed-Forward Neural Network
ROC	Receiver Operating Characteristic
TPR	True Positive Rate
FPR	False Positive Rate
AUC	Area Under the Curve
GPU	Graphics Processing Unit
CPU	Central Processing Unit
AI	Artificial Intelligence
AJAX	Asynchronous JavaScript and XML
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
StepLrUpdater	Step Learning rate Updater
SGD	Stochastic Gradient Descent
AdamW	Adaptive Moment Estimation

Glossary

- **Deep Learning**: A subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Deep learning models simulate human decision-making by layering algorithms to create an artificial neural network that can learn and make intelligent decisions on its own.**[12]**

- **Fashion Item Recognition**: The process of using computer vision and deep learning technologies to automatically detect, identify, and classify different fashion items from images. This technique allows for the automation of cataloging fashion items and enhances search functionalities in retail applications.**[8]**

- **Flask-Web Application Development**: Refers to the creation of web applications using Flask, a lightweight and modular web framework for Python. Flask provides tools, libraries, and technologies that enable developers to build a web application. This framework is particularly favored for projects requiring a simple, yet scalable and robust, web application.

- **ResNet (Residual Network)**: A type of convolutional neural network (CNN) that includes "skip connections" or "shortcuts" to jump over some layers. ResNet models are widely used for image recognition tasks because they allow for training significantly deeper networks by addressing issues like vanishing gradients.**[13]**

- **AlexNet**: A deep convolutional neural network that was influential in demonstrating the capability of CNNs on large-scale image recognition tasks. AlexNet was the winner of the 2012 ImageNet Large Scale Visual Recognition Challenge by a substantial margin.**[16]**

- **ConvNeXt**: A modern convolutional neural network architecture that adapts elements from Transformer models, designed for high performance in vision-related tasks. It brings improvements in scalability and efficiency over traditional CNNs.**[17]**

- **DenseNet** : A convolutional neural network where each layer is connected directly to every other layer in a feed-forward fashion. DenseNets are particularly efficient and effective due to their dense connections leading to reduced parameter counts and improved accuracy.**[18]**

- **EfficientNet**: A family of convolutional neural networks designed to achieve state-of-the-art accuracy with an optimized number of parameters to be as efficient as possible. EfficientNet models scale uniformly at the level of depth, width, and resolution.**[19]**

- **MobileNet**: A class of efficient models for mobile and edge devices, designed for low computational resource consumption. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build lightweight deep neural networks.**[20]**

- **Vision Transformer**: A model architecture that adapts the Transformer, traditionally used for natural language processing, for image recognition tasks. ViTs divide an image into patches and process sequences of these patches, demonstrating excellent performance on large-scale image datasets.**[21]**

.

Chapter 1 Introduction

1.1 Background

In recent years, the fashion industry has seen a significant transformation through the integration of advanced technologies**[1]**. It is fashion item recognition that uses computer vision technologies, especially deep learning models to detect, identify, and classify fashion items within images**[2]**. Deep learning models, particularly CNN**[3]**, have proven highly effective in handling complex visual data and extracting intricate patterns that are beyond human capabilities. These models facilitate the automatic tagging and sorting of fashion items, reducing the need for manual input, which can be error-prone and time-consuming. This enhancement in automation not only streamlines inventory management but also optimizes the consumer shopping experience by providing more accurate search results and recommendations**[4]**.

But there is primary challenge which is the analysis of a vast and varied collection of fashion product designs, especially for large companies like Adidas AG™ that source designs from multiple global teams**[5]**. Current models, although improved with CNN and computer vision techniques, still struggle with the vast diversity and nuanced differences within fashion items such as clothing, accessories, and footwear**[6]**. These limitations lead continuous improvements in model architecture, feature extraction, and the training process to enhance classification accuracy and functionality in practical applications.

The other challenge in utilizing image recognition technologies for fashion applications lies in selecting the optimal deep learning models and techniques for feature extraction and classification. Determining the most effective model for achieving high accuracy remains difficult**[7]**. These challenges are compounded by the need to cater to a diverse range of styles and presentations within the fashion industry, making it essential to continuously refine and adapt these models to handle the complexity and variability of fashion items effectively.

The subsequent sections of this project report are structured as follows: Chapter 2 provides an overview of the related literature and delineates the principal attributes of each identified method in fashion item recognition using deep learning techniques. Chapter 3 delves into the specifics of the deep learning models employed, including their architectural nuances and the process of model training and validation using the

Fashion item dataset. The results of the experiments, highlighting the effectiveness of these models in classifying fashion items, are detailed in Chapter 4. Chapter 5 addresses the managerial and professional considerations of the project, discussing the ethical, logistical, and technical challenges encountered. The final chapter, Chapter 6, offers a comprehensive summary of the project, reflecting on its successes and discussing the inherent limitations and potential areas for future research.

1.2 Aim

This project aims to use different deep learning models to conduct fashion item recognition accurately , comparative analysis of prediction results from each models, and develop a flask-based web application to conduct fashion item recognition using deep learning with the models.

1.3 Objectives

a. Background research: Research on the existing product of fashion item recognition using deep learning. Study the relevant models that would be used in fashion item recognition.

b. Dataset construction: Construct the datasets which are related to fashion item images and processing the dataset

c. Model development: Implement the deep learning models. Train and test the deep learning models.

d. Model evaluation: Evaluate each trained models and save the trained models with the best prediction accuracy for further using

e. Web application development: Design the user interface prototype and function of the website. Develop the front-end following the prototype and back-end following the function.Integrated front-end and back-end.

f. Testing and debugging: Test the function of the website, if there is any bug, debug the website and solving the potential problems.

1.4 Project Overview

1.4.1 Scope

The main scope of this project includes the development of an advanced fashion item recognition system with the accuracy and reliability of current market offerings by

harnessing the capabilities of deep learning. Focused on innovating within the fashion industry's digital sphere, the project aims to provide users with sophisticated tools for precise and efficient identification of fashion items. This involves comprehensive research on existing recognition solutions, the meticulous construction and preprocessing of a diverse fashion image dataset, and the deployment of various deep learning model architectures for optimal performance. Alongside technical development, the project is dedicated to creating a user-friendly, flask-based web application, enabling seamless user interaction and real-time fashion item prediction. The initiative concludes with thorough testing and refinement of the web application to ensure reliability and ease of use, positioning it as a valuable contribution to the digital transformation of the fashion sector.

1.4.2 Audience

The audience for this project spans a diverse range of stakeholders who would be benefit from fashion item recognition technology using deep learning. The first type of audience is Fashion industry professional, including designers, stylists, retailers, and marketers, they will benefit from greater accuracy and efficiency in tasks like inventory management and trend analysis. The second type of audience is Fashion consumer who will enjoy more reliable tools for inspiration and shopping, enhancing their ability to explore and discover fashion items easily. The third type of audience is Researcher in computer vision and machine learning , they will find valuable insights and methodologies that advance the field and support further innovation. The forth type of audience is Technology developer, they will gain practical applications and strategies for implementing machine learning solutions in real-world scenarios.

Chapter 2 Background Review

A. Fashion Item Recognition

Fashion item recognition is the main topic of this project. According to Rajalekshmi**[8]**, fashion item recognition involves using computer vision and deep learning to identify clothing items such as shirts, dresses, and jeans, along with their colors, in images. Image recognition technology has become integral to the fashion industry, offering innovative solutions such as visual search, smart recommendations, virtual try-on experiences, and styling tools. By using computer vision algorithms, fashion brands and retailers enhance user experiences, streamline processes, and drive business growth. From enabling seamless online shopping with visual search capabilities to providing personalized recommendations based on individual preferences, image recognition is transforming how consumers discover, interact with, and purchase fashion items**[9]**.There are some challenges in the area of fashion items recognition , the task of identifying attributes of fashion clothing involves accurately capturing various features from images such as color, pattern, style, and material. The diverse appearances and styles of fashion garments, coupled with varying definitions and aesthetic standards among individuals, render the precise identification of these attributes challenging. For instance, the same garment may possess different attributes depending on the occasion or cultural context, adding complexity to attribute recognition. And fashion clothing retrieval entails finding other fashion items similar to a given clothing image. This necessitates models to understand the semantic and visual similarities between images and effectively compare and match different clothing items. Additionally, due to the diverse appearances and styles of fashion garments, as well as potential variations and noise between images, conducting effective fashion clothing retrieval poses a challenging task**[10]**.

B. Deep learning Models

Deep learning is a specialized branch of machine learning and artificial intelligence that is recognized as a key technology in the current Fourth Industrial Revolution, often referred to as Industry 4.0. It involves training artificial neural networks on large sets of data, enabling them to learn from this data autonomously**[11]**. The core technique of Deep learning is Deep learning model which is a type of computational model that utilizes multiple layers of processing to learn data representations at various levels of abstraction. Deep learning methods have significantly enhanced performance in various

domains, including speech recognition, and object detection like fashion item detection, as well as in fields like drug discovery and genomics**[12]**. Here are seven deep learning models which will involve in this project. The first model is Resnet**[13]** which is a type of deep neural network known for its architecture that utilizes "skip connections" or "shortcuts" to jump over some layers, significantly improving the ability to train deeper networks by addressing the vanishing gradient problem. According to the paper of Shafiq**[14]**, Residual Network allows the output of the current layer add the input of the previous layer which means that the network could learn more easily and get results in higher performance. And from the study of Ebrahimi**[15]**, deep Residual Network could capture the information more accurately and conduct the training process more quickly than other equivalent neural network which don't equipment the residual connections. The second model is AlexNet**[16]** that is renowned as a milestone in CNN for image classification. Its design, includes convolutional layers, pooling, dropout, GPU acceleration, parallel computing, and ReLU activation functions, has become the industry standard in computer vision. One unique advantage of AlexNet is its direct input of images into the classification model. And the convolutional layers automatically extract image edges, while fully connected layers learn from these features. The addition of more convolutional layers theoretically enables effective extraction of complex visual patterns.The third model is the ConvNeXt**[17]** which is a modern convolutional neural network architecture that improves based on traditional CNN designs by incorporating innovations from transformer models, optimizing it for better performance and scalability in various vision tasks.The forth model, Dense Convolutional Network**[18]**, is a also type of convolutional neural network that is distinct because of its densely connected layers.The main innovation of DenseNet lies in its architecture, where each layer is directly connected to every other layer in a feed-forward fashion.And the key component of DenseNet is the "Dense Block," where each layer is connected to every other layer in a feed-forward fashion. Between these dense blocks, transition layers are used for down-sampling the feature maps to reduce their size and to manage the model complexity.The fifth model is MobileNet**[19]** which is a breakthrough in the realm of computer vision, offering a class of efficient models that cater specifically to mobile and embedded vision applications. Characterized by their lightweight architecture, these models significantly reduce computational costs and model sizes through the ingenious use of Depthwise separable convolutions. This approach not only makes MobileNets exceptionally efficient but also allows for a flexible

trade-off between latency and accuracy, thanks to two global hyperparameters. Despite their compactness, MobileNets deliver robust performance across a broad spectrum of applications, from ImageNet classification to object detection and beyond, achieving near state-of-the-art accuracy.The sixth model is the EfficientNet**[20]** which stands out in the landscape of Convolutional Neural Networks due to its innovative scaling method that uniformly increases network depth, width, and resolution with a fixed set of scaling coefficients. This balanced scaling approach allows EfficientNet to achieve significantly higher accuracy and efficiency on various benchmark datasets compared to other state-of-the-art CNN architectures. By optimizing these three dimensions together, EfficientNet can efficiently utilize computational resources, resulting in models that are not only smaller in size but also faster in inference while maintaining or even surpassing the accuracy of much larger models.The last model is Vision Transformer**[21]** The Vision Transformer represents a significant shift in the approach to computer vision problems, leveraging the architecture commonly used in processing sequential data (like text) for image classification tasks.

Chapter 3 Methodology

3.1 Approach

3.1.1 DataSet

In order to achieve the aim of the project, The datasets which is related to fashion item was found on Kaggle website. The name of the data set is “Fashion Product Image Dataset” which include 44446 product images and 142 product classes. All the images in this data set have the same resolution and the same background which is white and clean showed in figure 1

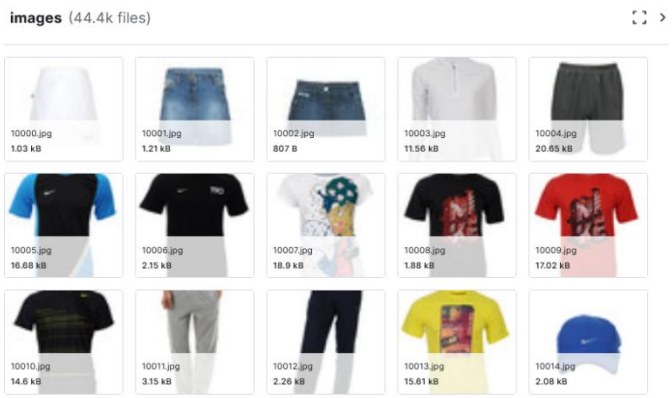

Figure 1 Some images in the data set

There is also a file called “styles.csv” which stores the details of all the fashion product images in the dataset showed in figure 2. The column which is called “article type” is the class of product. And the name of each images in the dataset is the product id which is matched the class also called article type in “styles.csv”.

styles.csv (4.33 MB)

Detail | Compact | Column

id	gender	masterCat...	subCateg...	articleType
15970	Men	Apparel	Topwear	Shirts
39386	Men	Apparel	Bottomwear	Jeans
59263	Women	Accessories	Watches	Watches
21379	Men	Apparel	Bottomwear	Track Pants

Figure 2 Some information in styles.csv

3.1.2 Fashion item recognition Models Training

The models will load the pre-trained weight from Pytorch, and after the model modification, the models are trained using the train and test datasets. The epoch for each training is 300, the system evaluate the model after each epoch and save the checkpoint with the highest accuracy which will be used to conduct fashion item recognition.Figure 3 shows the system architecture diagram of the project.

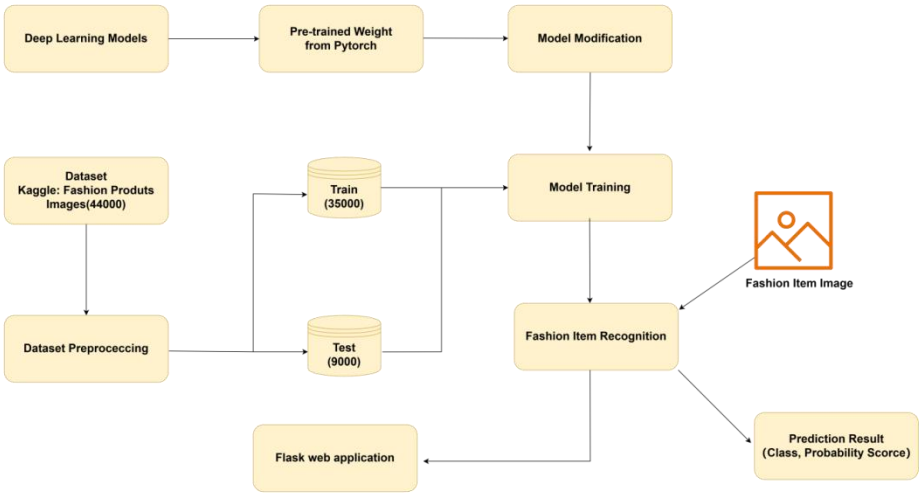

Figure 3 System Architecture Diagram

3.1.3 Model Deployment

To deploy a trained model for predicting fashion items using a Flask web application, a user-friendly interface is developed, integrating HTML, CSS, and JavaScript. This interface allows users to select a model and upload an image of the fashion item via a form. When the form is submitted, a POST request is made to the Flask back-end. Flask handles this request, where the image is processed using Python's Pillow library to prepare it for prediction. The chosen model, loaded the trained model, then processes the image to predict the fashion item's characteristics. The prediction results are formatted and returned to the front-end via a Flask response. This response is managed through AJAX calls within the JavaScript code, ensuring that the results are dynamically displayed to the user without reloading the page. This entire process ensures a seamless interaction, efficient server-side processing, and a dynamic user experience with minimal delays.

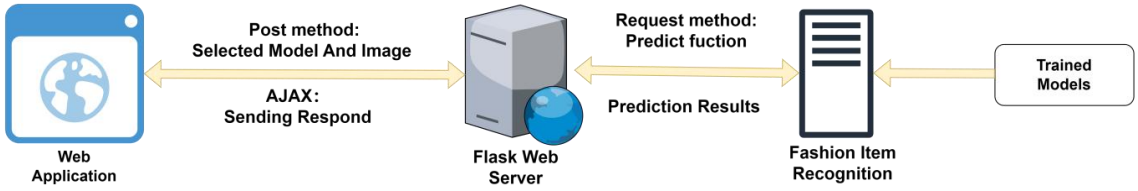

Figure 4 Models deployment using flask web application

3.1.4 The deep learning models

The seven deep learning models will involve in this project to conduct fashion item recognition are Resnet，AlexNet, ConveNext, DenseNet, EfficientNet, MobileNet and Vision Transformer. Here are the explanations of each deep learning model.

A. Resnet 50

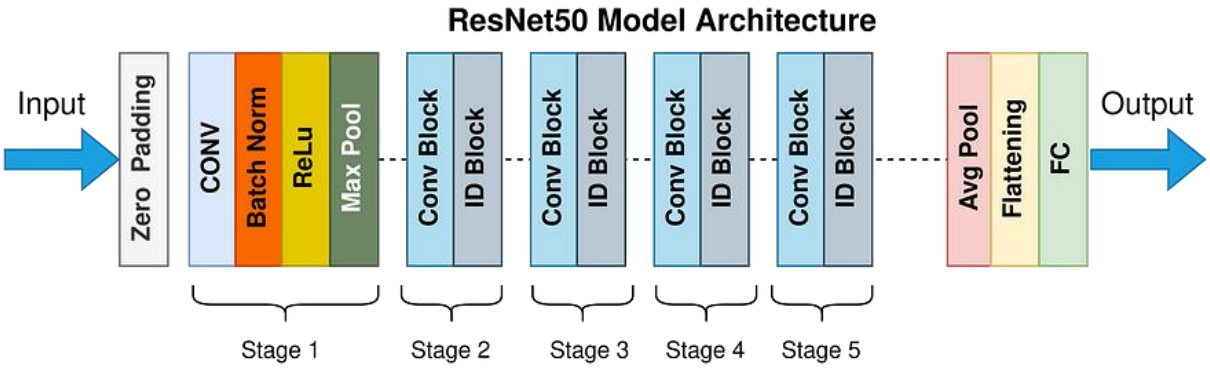

Figure 5 ResNet50 Model Architecture (Mukherjee, 2022)

ResNet50 is a variant of the ResNet architecture that uses 50 layers deep, designed to solve the problem of vanishing gradients and enable the training of deeper neural networks more effectively. The core idea behind ResNet50, and ResNets in general, is the use of "residual blocks." Each block contains a few layers that perform a series of convolutions and then add their output to the input of the block, essentially allowing the network to learn an identity function if necessary, which prevents the gradient from vanishing as the network depth increase .

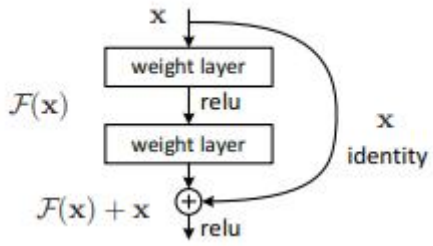

Figure 6 skip connections in Resnet (He, 2015)

A typical residual block consists of a few convolutional layers, each followed by a batch normalization layer and a ReLU activation function.

Here is the output equation of Residual Block:

$$y = F(x, \{W_i\}) + x \quad (1)$$

Where x is the input feature, $F(x, \{W_i\})$ represents the output from a series of convolutional layers, $\{W_i\}$ are the weights of the convolutional layers, y is the output of the residual block

The formula of ReLU is given as below:

$$F(x) = max(0, x) \quad (2)$$

This means that for any input x, the ReLU function outputs x if x is greater than zero, and outputs zero otherwise.

The formula of Batch Normalization is given as below:

$$y = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (3)$$

x is the input feature, μ_B and σ_B^2 are the mean and variance of the batch, ϵ is a small constant (to prevent division by zero).

B. AlexNet

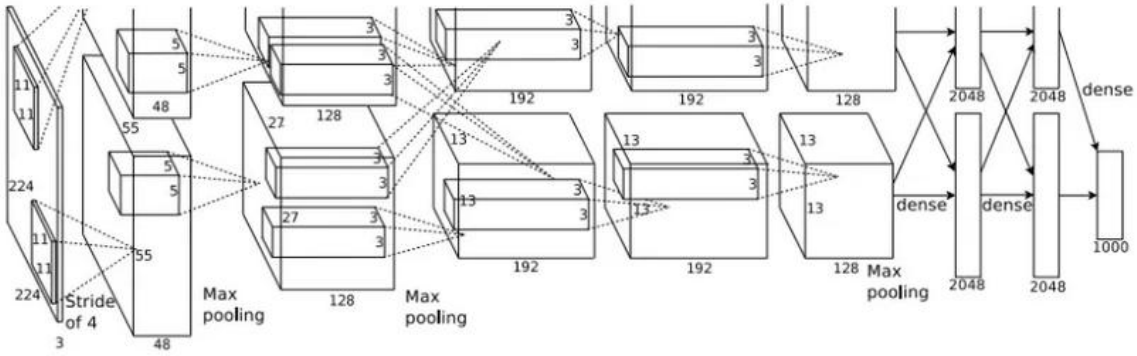

Figure 7 AlexNet Architecture (Pujara 2020)

AlexNet is a deep convolutional neural network architecture comprising 5 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer. Each convolutional layer employs ReLU activation functions, while max-pooling layers perform max pooling. The input size, often cited as 224x224x3, effectively becomes 227x227x3 due to padding. The presence of fully connected layers fixes the input size.

AlexNet uses multiple convolution layers to extract features from images using this operation.Here is the output equation of Convolution Operation:

$$O(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} I(x + i, y + j) \cdot W(i, j) \quad (4)$$

Where O is the output feature map, I is the input feature map,W is the convolution kernel, x, y are spatial coordinates, with k being the range of the kernel size.

Max pooling is used to reduce the size of the feature maps while retaining important features. The formula of Max pooling is given as below:

$$O(x, y) = \max_{(i, j) \in R(x, y)} I(i, j) \quad (5)$$

$R(x, y)$ is the pooling window centered at (x, y), I is the input feature map, and O is the output feature map

C. ConveNext

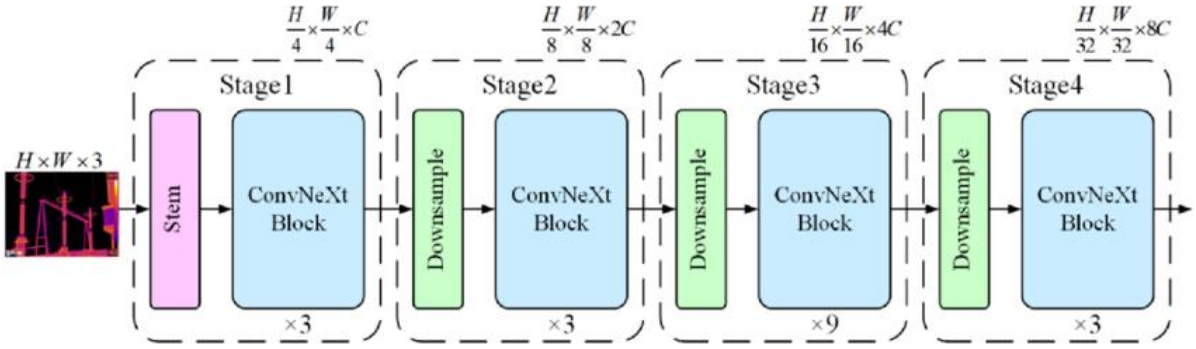

Figure 8 ConvNeXt architectures (Liu 2020)

The ConvNeXt architecture consists of a refined arrangement of layers and blocks that collectively aim to optimize deep learning performance for vision tasks. Starting with a stem layer, ConvNeXt processes input images through a single, large-kernel convolution to efficiently reduce spatial dimensions. The core of the architecture comprises multiple stages of ConvNeXt blocks, each featuring depthwise separable convolutions for effective spatial processing with reduced computational demands. These blocks incorporate layer normalization prior to convolution, a shift from traditional ConvNets, and utilize the GELU activation function post-convolution for smoother non-linearity.

ConvNeXt utilizes standard convolutional operations but adapts the design principles from Transformers. The basic operation in a convolutional layer can be described by the formula:

$$Y = W * X + b \tag{6}$$

Here, Y is the output, W represents the weights of the convolutional filters, X is the input feature map, b is the bias, and $*$ represents the convolution operation.

ConvNeXt uses layer normalization. Layer normalization can be represented as:

$$LN(x) = \frac{x-\mu}{\sigma} \cdot \gamma + \beta \tag{7}$$

Where x is the input feature, μ and σ are the mean and standard deviation calculated across the channel dimensions, γ and β are learnable parameters for scaling and shifting

ConvNeXt adopts the GELU as the activation function. The GELU activation function can be represented as:

$$\mathrm{GELU(x)} = \mathrm{x} \cdot \Phi(x) \tag{8}$$

$\Phi(x)$ is the cumulative distribution function of the standard Gaussian

D. DenseNet

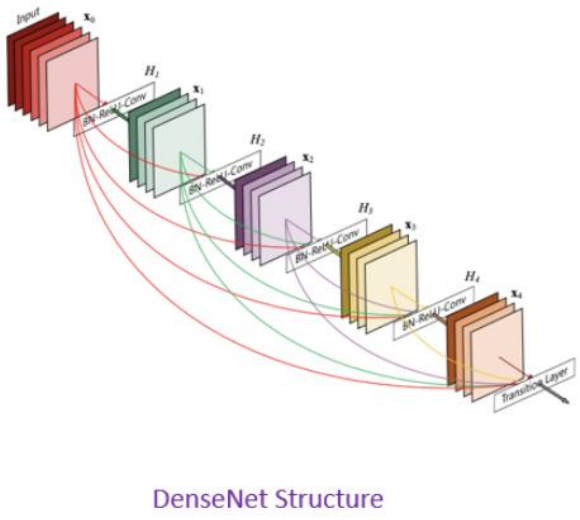

Figure 9 DenseNet Architecture (Huang 2018)

DenseNet is a convolutional neural network architecture where each layer is directly connected to every other layer in a feed-forward fashion. Unlike traditional CNNs where the output from one layer is only sent to the next layer, DenseNet layers each receive additional inputs from all preceding layers and pass on their own feature-maps to all subsequent layers.

In DenseNet , the core feature is that each layer is directly connected to every other layer that comes before it. This architecture is typically expressed mathematically as follows:

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]) \tag{9}$$

Here x_l represents the output of the l-th layer, $[x_0, x_1, \ldots, x_{l-1}]$ denotes the concatenation of the outputs from all preceding layers from 0 to $\mathrm{l}-1$. and H_l is a composite function applied at the l-th layer which comprising BN(3), a ReLU(2) activation function, and a convolution operation(6).

E. MobileNet

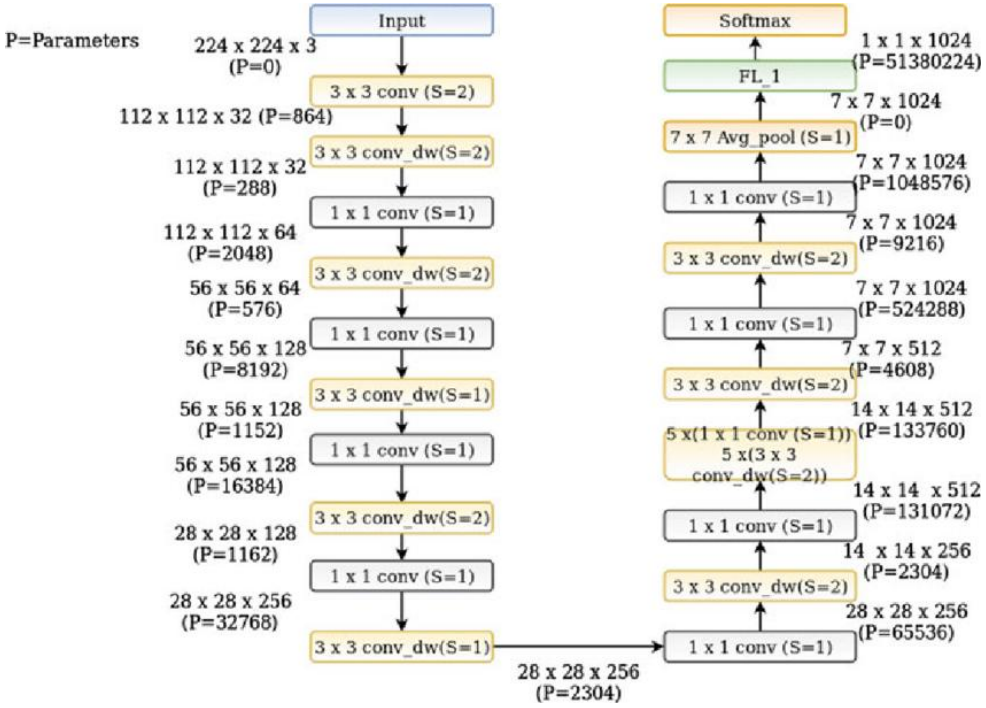

Figure 10 architecture of the MobileNet (Radhika, 2020)

MobileNet architecture consists of a stack of depthwise separable convolution layers where each layer is a combination of a depthwise convolution and a pointwise convolution. The depthwise convolution applies a single filter to each input channel, and the pointwise convolution, a 1x1 convolution, then combines the outputs of the depthwise convolution across channels. This setup not only reduces the computational

load but also integrates the extraction of features and the creation of new feature channels.

The first part is Depthwise convolution with each filter applied per input channel. This means each filter operates on a single channel.The formula is followed:

$$G_{k,l,m} = \sum_{i,j} K_{i,j,m} \cdot F_{k+i,\, l+j,\, m} \tag{10}$$

Here, F is the input feature map, K is the convolutional kernel, G is the output feature map, m represents the channel index, and i and j are spatial indices of the kernel.

The second part is pointwise convolution, also known as 1×1 convolution, is used to combine and re-project the channels outputted by the depthwise convolution. It operates on the output of the depthwise convolution to combine features across channels into a new set of feature maps. The formula is:

$$H_{k,\,l,\,n} = \sum_{m} W_{m,\,n} \cdot G_{k,\,l,\,m} \tag{11}$$

Where W is the 1×1convolutional kernel, H is the final output feature map, and n is the output channel index.

F. EfficientNet

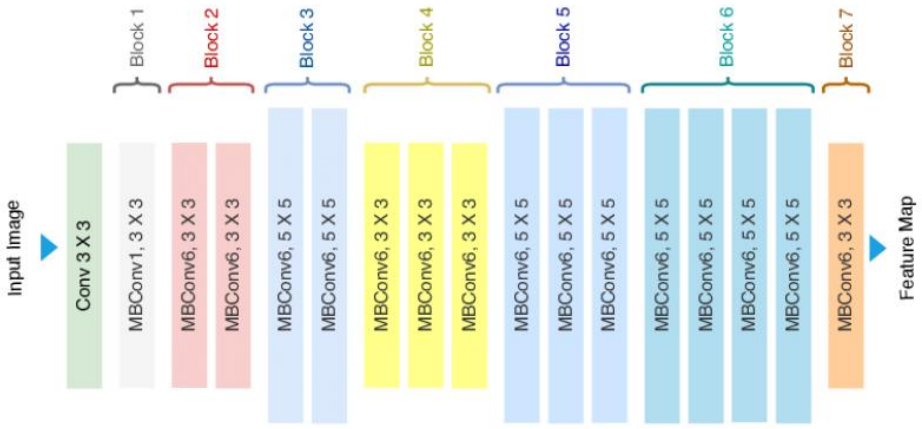

Figure 11 EfficienNet Architecture (Ahmed,2021)

EfficientNet uses the MBConv block, an architectural component based on the inverted residual structure from MobileNetV2, incorporating depthwise separable convolutions which contains the formula 10 and formula 11. An MBConv block typically features one or more 1×1 convolutions to alter channel dimensions, depthwise convolutions to filter features, possible dropout layers, and skip connections.

G. Vision Transformer

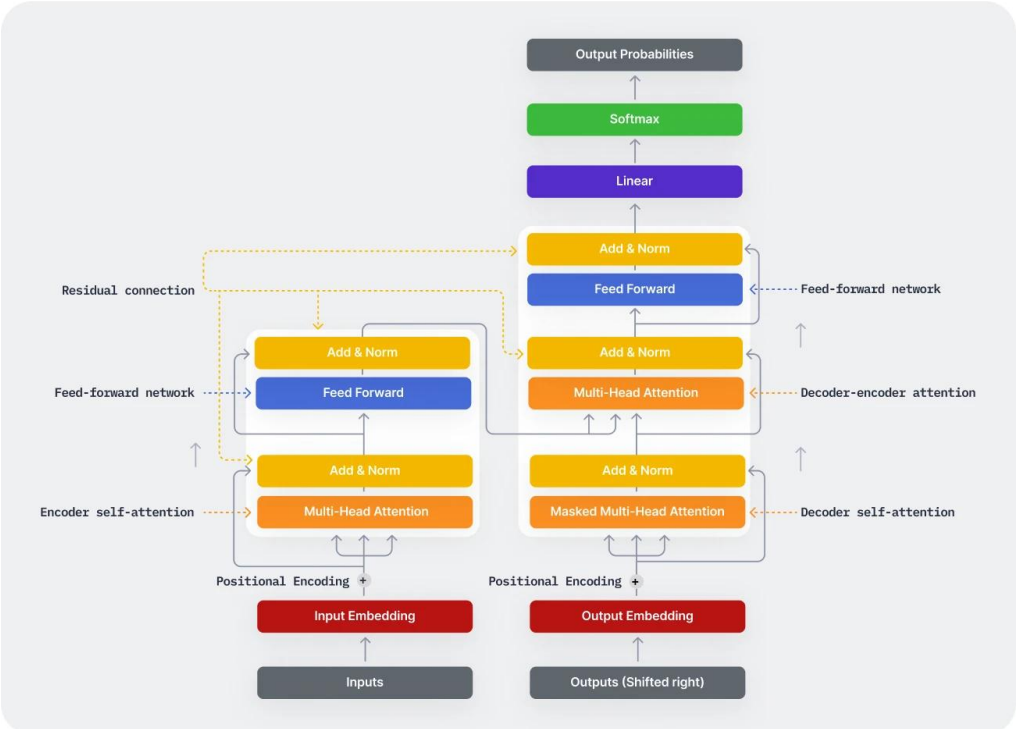

Figure 12 Vision Transformer (Shah 2022)

The ViT processes images through a sequence of distinct stages, starting with the division of an image into fixed-size patches that are linearly embedded and augmented with positional embeddings to preserve spatial information. It then utilizes a Transformer Encoder, which applies a Self-Attention Mechanism to weigh the importance of different patches relative to each other and processes these through a Feed-Forward Neural Network. Finally, a Classification Head, typically a linear layer followed by a softmax

function, predicts class probabilities. These components enable ViT to effectively understand and classify complex visual data by learning intricate patterns and relationships across image segments.

The formula for the patch embeddings can be expressed as:

$$x_p = [z_1^P; z_2^P; \ldots; z_N^P]E \tag{12}$$

where z_i^P represents the flattened patch, and E is a trainable embedding matrix that maps the flattened patches to a higher dimensional space.

The core of ViT is a series of Transformer layers, each comprising two sub-layer:

The first one is Self-Attention Mechanism which can be presented as:

$$SA = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{13}$$

Here, Q, K, and V represent the query, key, and value matrices derived from the input embeddings, and $\mathrm{d_k}$ is the dimensionality of the keys and queries, facilitating scaling in the softmax function.

The second one is Feed-Forward Neural Network which can be presented as:

$$\mathrm{FFN(x)} = \max(0, \mathrm{xW_1} + \mathrm{b_1})\mathrm{W_2} + \mathrm{b_2} \tag{14}$$

Each layer has its own set of weights and biases ($\mathrm{W_1}$, $\mathrm{b_1}$, $\mathrm{W_2}$, $\mathrm{b_2}$), and applies a ReLU activation function after the first transformation.

The softmax function could be expressed as:

$$\mathrm{softmax(Z_i)} = \frac{\mathrm{e^{z_i}}}{\sum_{\mathrm{j=1}}^{\mathrm{K}} \mathrm{e^{z_j}}} \tag{15}$$

$\boldsymbol{Z_i}$ is the logit calculated by formula 6 for class i, $\boldsymbol{K}$ is the total number of classes, and $\boldsymbol{e}$ is the natural exponential function.

3.1.5 Evaluate Method

In order to conduct comparative analysis of the prediction results from the models. There are some methods would be used to Evaluate the results.

A. Accuracy

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Numer of Predictions}} \times 100 \quad (16)$$

The Accuracy evaluates the performance of a model by determining the percentage of predictions it got right. It does this by comparing the model's predicted labels with the actual, true labels of the data. The function counts how many predictions match the true labels and divides this number by the total number of predictions made.

B. Loss

$$\text{Average Loss} = \frac{1}{N}\sum_{i=1}^{N} Li \quad (17)$$

N is the total number of batches, Li is the loss for the i-th batch.

The average loss calculation in machine learning model evaluation aggregates the error across all batches of a dataset to measure overall model performance. By computing the loss for each batch, summing these losses, and dividing by the number of batches, we obtain a normalized metric that reflects the model's accuracy in predicting unseen data. This process is pivotal for assessing a model's effectiveness, guiding improvements, and comparing different models, making average loss a fundamental indicator of how closely a model's predictions align with actual targets.

C. Recall

$$\text{Recall} = \frac{\text{True Positives(TP)}}{\text{True Positives(TP)+False Negatives(FN)}} \quad (18)$$

Recall measures how well a classification model identifies all relevant instances of a particular class.The true positives represent the instances correctly predicted as belonging to the positive class. And the false negatives represent the Instances that belong to the positive class but were incorrectly predicted as not belonging to it. The recall is then computed as the ratio of true positives to the sum of true positives and false negatives.

D. F1-score

$$\text{F1 Score } = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{19}$$

The F1 score harmonizes these two by multiplying their product by 2 and then dividing by the sum of precision and recall. This creates a single metric that considers both the false positives (affecting precision) and false negatives (affecting recall). The F1 score is particularly useful when need to balance precision and recall, which is often the case when the costs of false positives and false negatives are different.

Where Precision is defined as:

$$\text{Precison } = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{20}$$

The precision function computes the precision metric for a classification model, focusing on a specific class designated as the positive class. Precision represents the accuracy of the model in predicting positive instances and is calculated as the proportion of true positive predictions out of all positive predictions made by the model.The true positives(TP) are instances where the model correctly identifies the positive class, and the false positives(FP) are instances where the model incorrectly labels an instance as positive when it is actually negative.

E. Roc-curve

The Receiver Operating Characteristic (ROC) curve is a graphical representation used in binary classification to evaluate the performance of a classification model at various threshold settings. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold levels, providing insight into the trade-off between detecting true positives and falsely predicting negatives as positives.

The True Positive Rate(TPR) is the proportion of actual positive cases correctly identified by the model. It can be calculated as:

$$\text{TPR } = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{21}$$

Where TP is the number of true positives, and FN is the number of false negatives.

The False Positive Rate(FPR) is the proportion of actual negative cases that were incorrectly classified as positive by the model. It is calculated as:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} \tag{22}$$

Where FP is the number of false positives, and TN is the number of true negatives. The ROC curve's area under the curve (AUC) provides a single measure of a model's overall performance. An AUC of 1.0 represents a perfect model; an AUC of 0.5 suggests a model with no discriminative ability, equivalent to random guessing. And a model whose ROC curve is closer to the top-left corner indicates a better performance, with higher TPRs and lower FPRs. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.

F. Probability Score

Probability scores are the output of a softmax function applied to logits, which are the raw predictions made by a neural network's final linear layer. The logits represent unnormalized scores for each class that the model computes based on the input features. By applying the softmax function, these logits are transformed into probability scores, which normalize the raw scores into a probability distribution across all classes. Each probability score directly reflects the likelihood, as determined by the model, that the input belongs to a particular class. This transformation not only makes the model's outputs interpretable as probabilities but also ensures that these scores sum to one, thus providing a quantifiable measure of the model's confidence in each class prediction. The formula of softmax function is showed in formula16

3.2 Technology

Hardware	Software:
CPU: Intel core I5	System: Windows 11
GPU: RTX 2070 Ti 16GB	Development Language: Python 3.8.18, HTML,CSS, JavaScript, Flask
Memory: 16GB	CUDA: 11.8
	Pycharm Professional
	Pytorch 2.2.1
	Kaggle

Table 1 The Development hardware and software

The whole development of project is on a windows 11 system, utilizing a suite of software tailored for various tasks. Development languages include Python, HTML, CSS, and JavaScript, with Flask for building web applications. PyTorch is employed for deep learning tasks, set up within a Conda virtual environment on Anaconda. PyCharm Professional serves as the primary IDE for all coding activities. Kaggle is used for constructing datasets, essential for training deep learning models. Additionally, CUDA supports GPU acceleration, enhancing the training process of models directly on the computer using both GPU and CPU.

3.3 Project Version Management

To mitigate the risk of code loss, the code is regularly uploaded to a GitHub repository for backup purposes. This strategy ensures that all changes and progress on the projects are securely stored off-site and can be retrieved at any time. Using GitHub also facilitates version control, allowing for tracking of modifications and easy collaboration.

The repository is accessible at the URL: https://github.com/Davidyiiiii/Project

Chapter 4 Implementation and Results

4.1 The pre-process of Dataset

4.1.1 Classification and Splitting of Dataset

Due to the images in dataset were not classified and all of them are in a single folder, So I decided to use the id of image and the matched class in “styles.csv” to classify the images and put the same class images into a folder which is named as the class name. I also split the dataset into two parts which are train and test. The train to test ratio is 4:1.

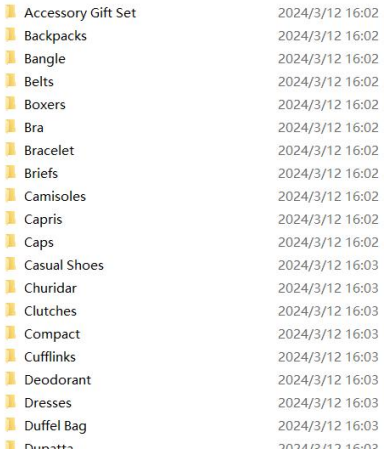

Figure 13 Some classifed folder of images

4.1.2 Cleaning the Dataset

There are some type of images with low number which means that it is not enough to conduct deep learning. In this case, I decided to clean the dataset by deleting the classes whose image number is lower than 25.

Figure 14 The original dataset

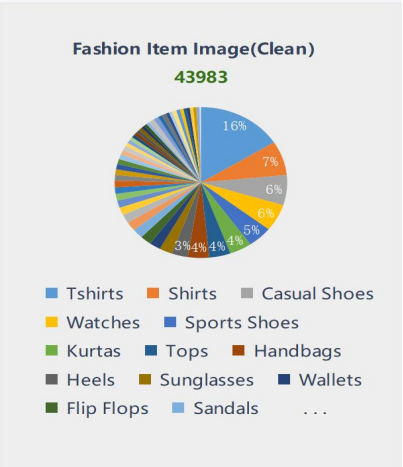

Figure 15 The cleaned dataset

After cleaning the dataset, the total quantity of fashion item images decreased from 44,446 to 43,983. And The total classes of products also decreased from 142 to 87.

4.1.3 Data Enrichment

A variety of data enrichment methods were employed to enhance the robustness and improve the generalization ability of the deep learning models used for fashion item recognition. These methods are designed to introduce variability into the training dataset, helping the models learn to recognize fashion items under diverse conditions and thereby reduce overfitting.

Here are the data enrichment methods in table 2.

Method	**Detail**
Random Resized Crop	Size = 96 X 96
Random Flip	Flip Prob:0.5, Direction: “horizontal”
Color Jitter	Brightness=0.4, Contrast=0.4, Saturation=0.4
Shear	±0.3° Horizontal, ±0.3° Vertical
Solar Size	Thr: from 0 up to 256
Cut Out	Shape: from 1 up to 41

Table 2 The Data Enrichment

4.2 Model Development

In this project, powerful weights of pre-trained models from PyTorch's torchvision library are used in the model development, which have been pre-trained on the extensive ImageNet dataset. This choice allowed to utilize the transfer learning approach, significantly reducing the training time and computational resources required. Each model, including ResNet, AlexNet, and others, was carefully selected based on its relevance to our task of fashion item recognition and was fine-tuned to our dataset, which consists of 87 unique fashion classes. The adaptation involved modifying the final classification layers to predict our specific set of classes and optimizing the models using an appropriate loss function and optimizer. The resulting models were then evaluated for accuracy, precision, recall, and F1-score on a validation set to ensure

robust performance. With satisfactory results, the models with best accuracy were saved, ready to be integrated into our web application for real-world testing and use.

Here are the hyper parameter configuration of each deep learning models:

Parameter	Value
Learning Rate	0.1
Epoch	300
Batch Size	256
Optimizer	SGD
Momentum	0.9
Weight Decay	1e-4
Number of classes	87
StepLrUpdater	(30, 60, 90)

Table 3 The hyper parameter of Resnet50 Model

Parameter	Value
Learning Rate	0.0008
Epoch	300
Batch Size	256
Optimizer	SGD
Momentum	0.9
Weight Decay	1e-4
Number of classes	87
StepLrUpdater	step=15

Table 4 The hyper parameter of AlexNet Model

Parameter	Value
Learning Rate	4e-3*128/64
Epoch	300
Batch Size	128
Optimizer	AdamW
Warm up Iters	20
Warm up Ratio	1e-3
Weight Decay	0.05
Number of classes	87

Table 5 The hyper parameter of ConvNeXt Model

Parameter	Value
Learning Rate	0.0008
Epoch	300
Batch Size	256
Optimizer	SGD
Momentum	0.9
Weight Decay	1e-4
Number of classes	87
StepLrUpdater	step=(30, 60, 90)

Table 6 The hyper parameter of DenseNet Model

Parameter	Value
Learning Rate	0.008
Epoch	300
Batch Size	256
Optimizer	RMSprop
Momentum	0.9
Weight Decay	1e-5
Number of classes	87
StepLrUpdater	step=(30, 60, 90), gamma=0.973
Dropout Rate	0.2

Table 7The hyper parameter of MoblieNet Model

Parameter	Value
Learning Rate	0.1
Epoch	300
Batch Size	256
Optimizer	SGD
Momentum	0.9
Weight Decay	1e-4
Number of classes	87
StepLrUpdater	step=(30, 60, 90)

Table 8 The hyper parameter of EfficientNet Model

Parameter	Value
Learning Rate	5e-4 * 32 / 64
Epoch	300
Batch Size	32
Optimizer	AdamW
Warm up Iters	20
Warm up Ratio	1e-3
Weight Decay	0.05
Number of classes	87

Table 9 The hyper parameter of Vision Transformer Model

4.3 The result of the model training

The comparative analysis of the training results systematically evaluates the performance of various deep learning models—AlexNet, ConvNeXt, DenseNet, EfficientNet, MobileNet, ResNet, and Vision Transformer on fashion item recognition. Each model is assessed over 300 epochs on metrics such as accuracy, loss, recall, precision, F1-score,on accuracy and loss curves and ROC curve, reflecting their ability to accurately classify fashion items.

Model	Evaluation method	Epoch 100	Epoch 200	Epoch 300
Alexnet	Accuracy	76.03	79.62	81.99
	Loss	0.78	0.65	0.57
	Recall	56.43	64.10	68.27
	Precision	67.42	73.02	76.34
	F1-score	58.23	66.20	70.32

Table 10 The evaluation matrix of Alexnet model

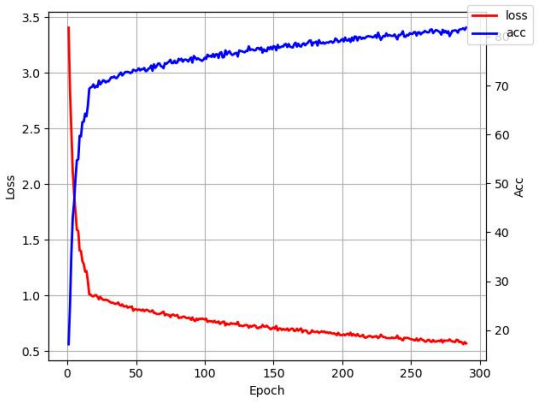

Figure 16 AlexNet accuracy and loss

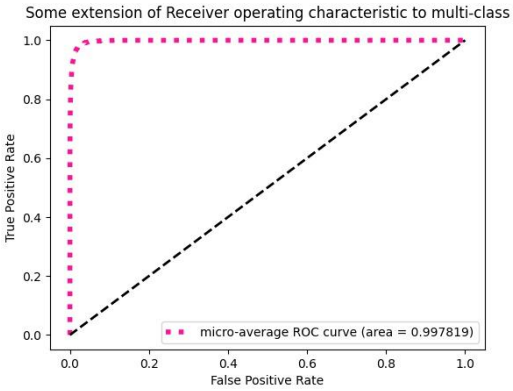

Figure 17 AlexNet Roc Curve

The training results of the AlexNet model demonstrate a steady improvement over time across multiple metrics.According to the accuracy and loss graph, loss(red line) starts high and decreases sharply within the first few epochs, which indicates that the model is quickly learning from the training data. After about 50 epochs, the loss continues to decline, but at a much slower rate, indicating the model is refining its understanding of the data. And the accuracy(blue line) climbs steeply initially, which suggests that the model is quickly improving at classifying the training data correctly. After around 50 epochs, the accuracy continues to increase but plateaus, suggesting diminishing returns on learning from the training data. By epoch 300, the accuracy has increased to just under 82%, and the loss has been reduced to 0.57, indicating that the model is making progressively more accurate predictions and is learning effectively from the training data. The recall, precision, and F1-score also show considerable improvements, with the F1-score reaching over 70%, suggesting a balanced performance between precision and recall. The ROC curve, with an AUC (0.997819) which close to 1, confirms the model's excellent capability to distinguish between different classes.

Model	Evaluation method	Epoch 100	Epoch 200	Epoch 300
Convnext	Accuracy	67.29	76.22	87.76
	Loss	1.12	0.77	0.39
	Recall	42.41	58.90	79.82
	Precision	56.46	70.19	87.36

	F1-score	43.49	61.70	82.89

Table 11 The evaluation matrix of Convnext model

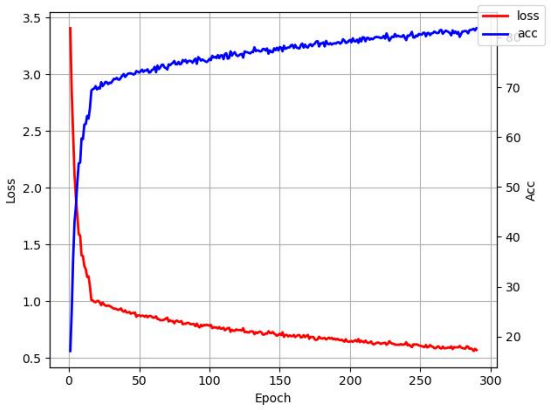

Figure 18 Convnext accuracy and loss

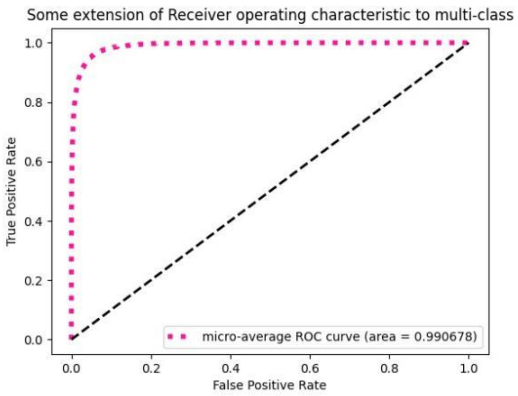

Figure 19 Convnext Roc Curve

The Convnext model demonstrates an exceptional trajectory of improvement across all metrics through the course of training, with accuracy soaring from 67.29% to 87.76% by epoch 300, and loss markedly decreasing from 1.12 to 0.39, indicating precise and reliable learning. Notably, recall and precision exhibit significant gains, suggesting enhanced model sensitivity and specificity, a sentiment echoed by the F1-score's rise to 82.89%. The loss (red line) shows a steep initial decline, indicating rapid learning, and then a consistent decrease, suggesting ongoing improvement without plateauing. The accuracy (blue line) complements this by showing a steady and significant rise, which does not appear to level off, indicating that the model might still benefit from further training beyond epoch 300.

Model	Evaluation method	Epoch 100	Epoch 200	Epoch 300
Densenet	Accuracy	75.30	75.58	75.63
	Loss	1.12	0.82	0.84
	Recall	42.41	40.02	40.40
	Precision	56.46	48.53	47.71
	F1-score	43.49	40.36	40.82

Table 12 The evaluation matrix of Densenet model

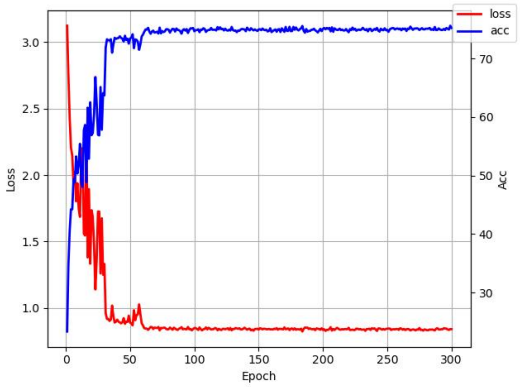

Figure 20 DenseNet accuracy and loss

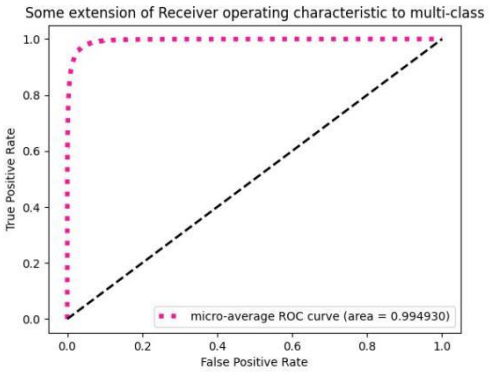

Figure 21 DenseNet Roc curve

The DenseNet model's training performance indicates a scenario where the accuracy slightly improves from 75.30% to 75.63% across 300 epochs, but the other metrics reflect a different story. The loss decreases from 1.12 to 0.82 at epoch 200 but then slightly increases to 0.84 at epoch 300, suggesting the model might be starting to overfit or is no longer gaining significant learning from the data. Interestingly, both recall and precision decrease over time, with recall dropping from 42.41% to 40.40%, and precision from 56.46% to 47.71%. This decline is mirrored in the F1-score, which decreases from 43.49% to 40.82%, indicating a deterioration in the model's balance between precision and recall. Graphically, the loss curve's upward trend after epoch 200 and the plateauing of accuracy suggest that the model is not benefiting from further training. The ROC curve, if assuming a similar pattern to the provided Convnext's ROC curve, with an AUC of 0.994930, which shows the model has strong classification power.

Model	Evaluation method	Epoch 100	Epoch 200	Epoch 300
Efficientnet	Accuracy	82.70	83.02	83.18
	Loss	0.76	0.91	0.93
	Recall	66.87	67.89	69.64
	Precision	75.56	77.68	77.26
	F1-score	69.87	70.40	71.60

Table 13 The evaluation matrix of EfficientNet model

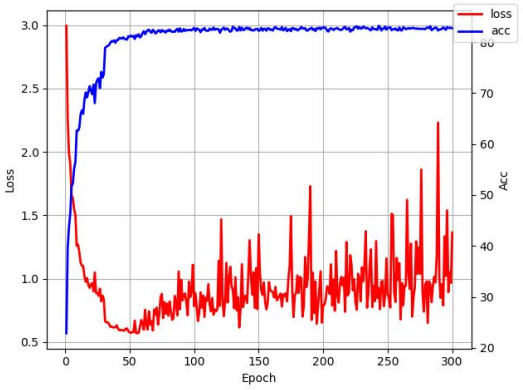

Figure 22 EfficientNet accuracy and loss

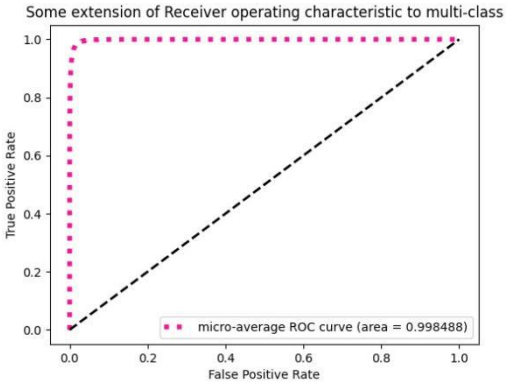

Figure 23 EfficientNet Roc curve

The EfficientNet model's performance indicates a subtle increase in accuracy from 82.70% to 83.18% over 300 epochs, suggesting a plateau in learning. Conversely, the loss has risen from 0.76 to 0.93, hinting at potential overfitting issues. The recall has positively increased from 66.87% to 69.64%, and the F1-score improved from 69.87% to 71.60%, both indicative of the model's growing capability to identify true positives and balance precision-recall trade-offs. Precision, while initially strong, shows a minor drop from epoch 200 to 300, from 77.68% to 77.26%. Graphically, the fluctuating loss spikes and stable high accuracy imply the model might benefit from optimization tweaks. The high AUC of 0.994488 suggests that the model is highly effective at class differentiation, which, combined with the other metrics, indicates a robust model.

Model	Evaluation method	Epoch 100	Epoch 200	Epoch 300
Mobilenet	Accuracy	81.05	85.50	87.42
	Loss	0.58	0.43	0.38
	Recall	66.30	76.35	80.39
	Precision	76.56	83.19	86.22
	F1-score	69.28	78.70	82.69

Table 14 The evaluation matrix of Mobilenet model

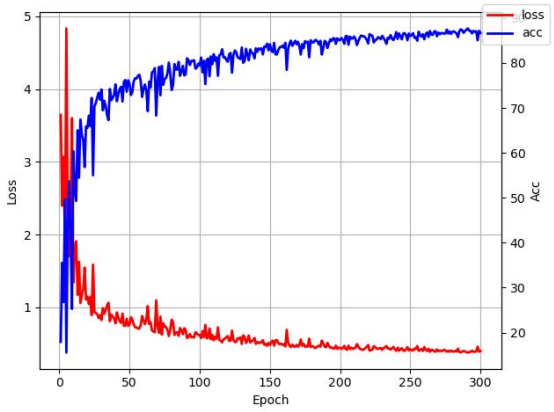

Figure 24 MoblieNet accuracy and loss

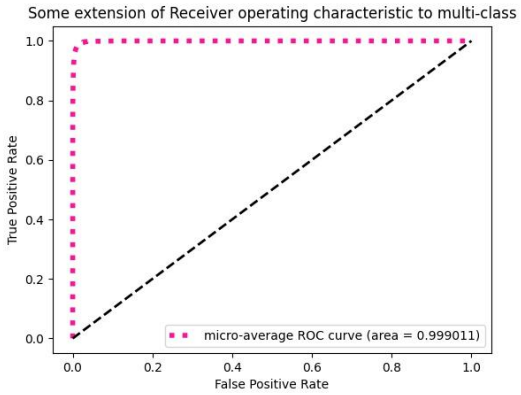

Figure 25 MoblieNet Roc curve

The MobileNet model showcases a robust learning curve, with accuracy improving significantly from 81.05% at epoch 100 to 87.42% at epoch 300, alongside a consistent decrease in loss from 0.58 to 0.38. This indicates the model is becoming increasingly effective at making correct predictions as training progresses. The recall improves from 66.30% to a notable 80.39%, suggesting the model is progressively better at identifying all relevant instances. Precision also sees a steady rise from 76.56% to 86.22%, highlighting a reduction in false positives. The F1-score, which takes into account both precision and recall, grows from 69.28% to 82.69%, reflecting a strong balance between the sensitivity and precision of the model. These metrics, along with the ROC curve's excellent AUC of 0.999011, underscore the model's robustness and its ability to discriminate effectively between classes.

Model	Evaluation method	Epoch 100	Epoch 200	Epoch 300
Resnet	Accuracy	78.64	78.98	79.13
	Loss	0.68	0.67	0.68
	Recall	57.74	59.66	59.51
	Precision	70.13	69.58	72.91
	F1-score	60.60	62.36	61.82

Table 15 The evaluation matrix of Resnet model

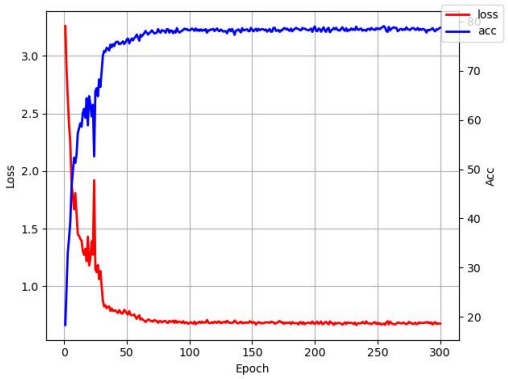

Figure 26 Resnet accuracy and loss

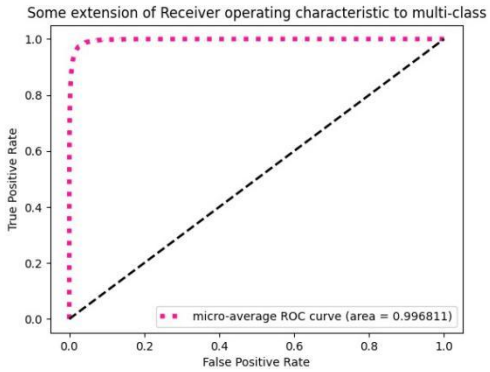

Figure 27 Resnet accuracy and loss

The ResNet model's training results exhibit a slight but steady improvement in accuracy, going from 78.64% at epoch 100 to 79.13% by epoch 300. Loss has seen little change, slightly decreasing to 0.67 at epoch 200 and then returning to 0.68 at epoch 300, suggesting the model may be approaching an optimal point in learning from the training data. Recall has a marginal increase from 57.74% to 59.66% and slightly dips to 59.51%, which could point to difficulties in capturing all true positives consistently. Precision experiences a slight fluctuation, dropping to 69.58% midway but recovering to 72.91%, indicating some variance in the model's ability to label positive instances correctly over time. The F1-score initially increases to 62.36% but then slightly falls to 61.82%, reflecting challenges in maintaining a balance between precision and recall. The ROC curve, with an AUC of 0.996811, indicates excellent discrimination between classes.

Model	Evaluation method	Epoch 100	Epoch 200	Epoch 300
Vision Transformer	Accuracy	70.15	80.96	86.38
	Loss	1.54	1.21	1.07
	Recall	47.97	68.00	78.50
	Precision	61.48	78.53	84.07
	F1-score	50.92	71.57	81.18

Table 16 The evaluation matrix of VIT model

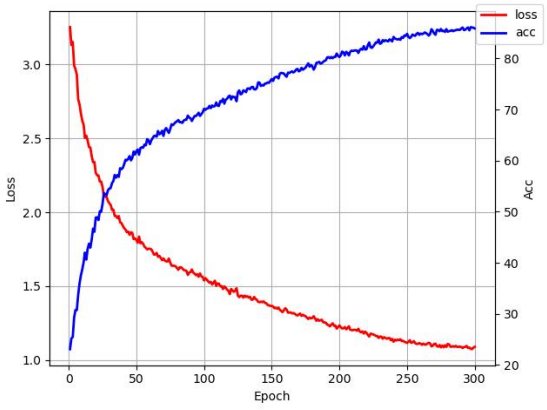

Figure 28 VIT accuracy and loss

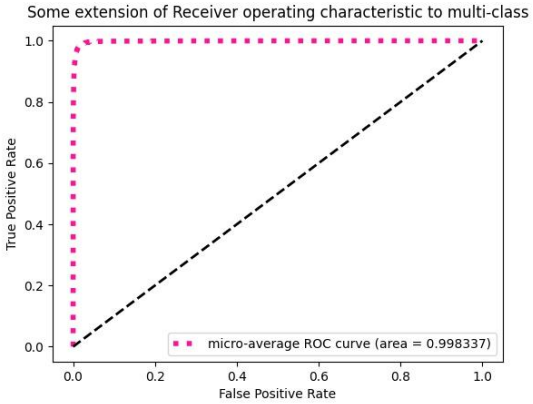

Figure 29 VIT Roc curve

The training results for the Vision Transformer model show a marked improvement across all key metrics over 300 epochs. Accuracy rises substantially from 70.15% to an impressive 86.38%, indicating the model's growing proficiency at correctly classifying images. Loss decreases significantly from 1.54 to 1.07, suggesting that the model is making fewer and fewer errors in its predictions as it learns. Recall, which quantifies the model's ability to correctly identify all positive instances, sees a significant increase from 47.97% to 78.50%, implying an enhanced capability to detect relevant cases. Precision, reflecting the accuracy of positive predictions, also jumps from 61.48% to 84.07%, pointing to fewer false positives. The F1-score, balancing precision and recall, climbs from 50.92% to 81.18%, suggesting a well-tuned model with a good balance between identifying true positives and avoiding false positives. The ROC curve, with an AUC of 0.998337, showing the model has excellent discriminative power between classes

The evaluation of the seven deep learning models for fashion item recognition demonstrates notable differences in performance across architectures. Models like MobileNet_v3 and Vision Transformer showed the best performance in accuracy, precision, and recall, indicating strong adaptability and effectiveness in classification tasks. ConvNeXt also performed well, showing significant gains in key metrics, which supports their robustness in handling complex image data. On the other hand, models like DenseNet and EfficientNet displayed some challenges, with DenseNet experiencing a decline in precision and recall, and EfficientNet showing signs of potential overfitting as indicated by an increase in loss over time.

4.4 Web application Development

For the development of front-end, the front-end of the web application serves as the interface where users interact with the application. It's crafted using HTML for structuring the page, CSS for styling to make the interface intuitive and visually appealing, and JavaScript for adding dynamic behaviors. Through the front-end, users can select a deep learning model, upload an image for recognition, and see the results—all within a single, seamless experience. JavaScript plays a pivotal role in ensuring that the interaction with the back-end is asynchronous, meaning the page doesn't need to be reloaded to show the prediction results, enhancing the user experience by making it smooth and responsive.

For the development of back-end, it built with Flask for this application, it's all about handling the requests sent from the front-end. It receives the image and the selected model from the user, processes the image using the chosen machine learning model, and computes the predictions. Each model corresponds to a specific route that dictates how the image is processed. After processing, the back-end assembles the prediction results into a JSON format and sends this data back to the front-end. This architecture allows for the heavy lifting of image processing and model prediction to be managed server-side, keeping the front-end light and focused on user interaction. Here is n example of using Resnet model to recognize a “belt” image.

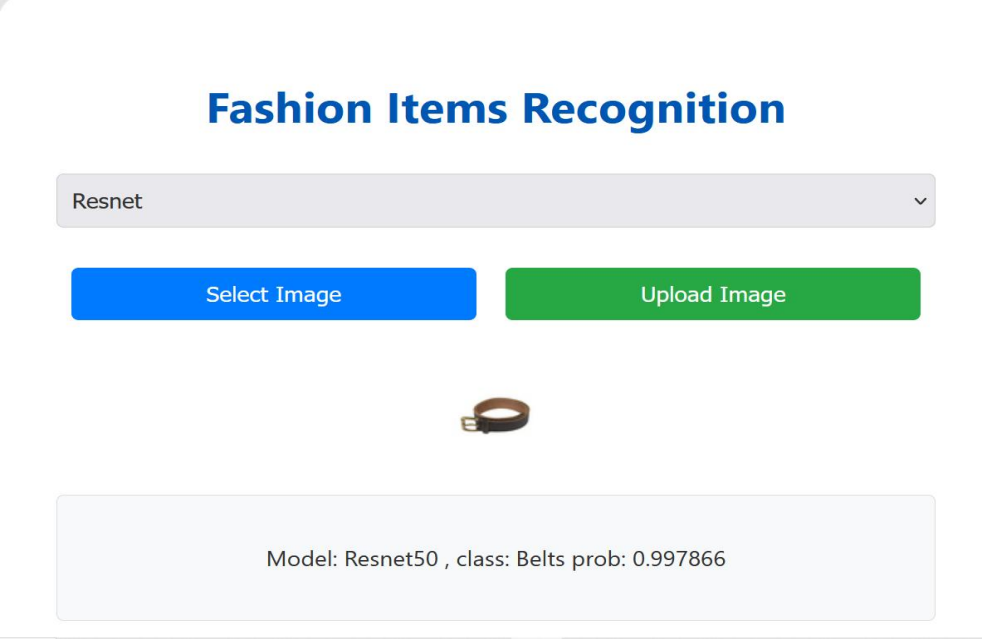

Figure 30 The model deployment using web application

4.5 Testing

4.5.1 Automatic Testing

This project employs Selenium IDE to automate the testing process for three primary functions of the web application, the tests of each function are following:

A. Submit without model

The first test is to verify the scenario where the user submits a fashion item for recognition without selecting a model. In this case, the web application should display the preview of the selected image. However, upon attempting to upload the image for prediction, the front-end should present a notification prompting the user to select a model and prevent the transmission of the image to the back-end.

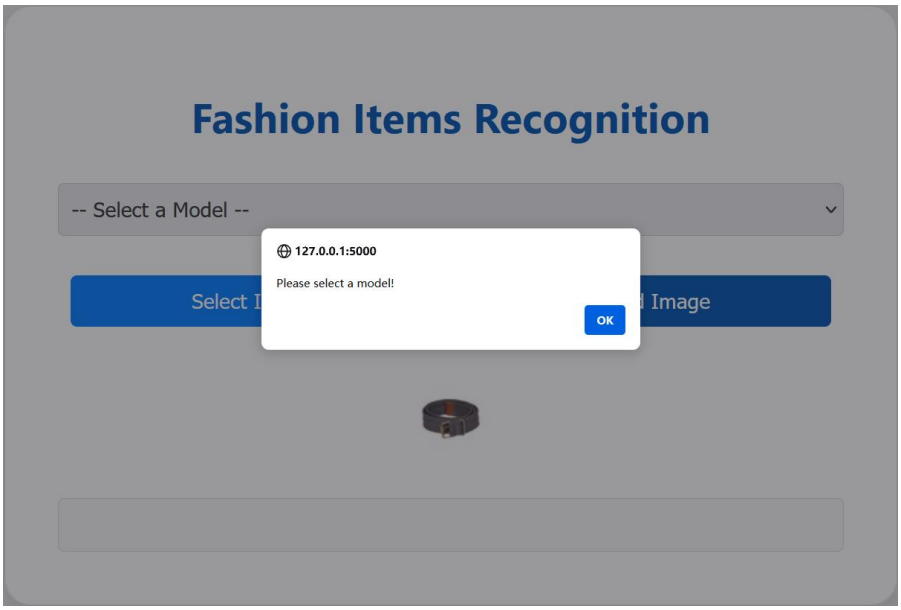

Figure 31 Submit without model

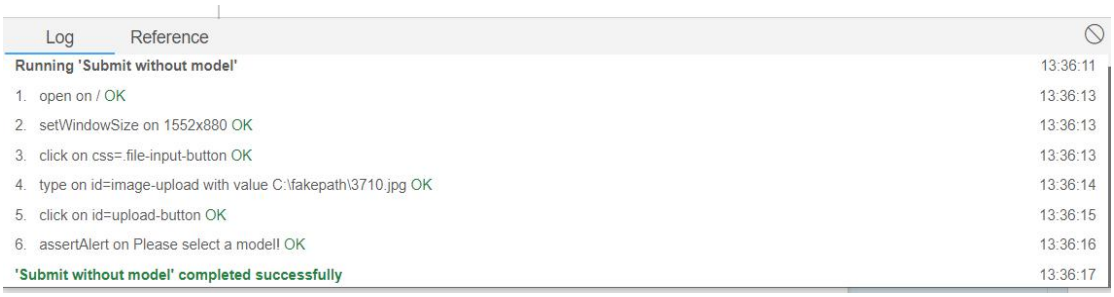

Figure 32 Test log of Submitting without model

B. Submit without image

The second test is designed to assess the scenario where the user selects a model for fashion item recognition but does not upload an image. In this case, the web application should display the text of the selected model. However, upon attempting to upload an image for prediction, the front-end should present a notification prompting the user to select an image and prevent the submission of the request to the back-end.

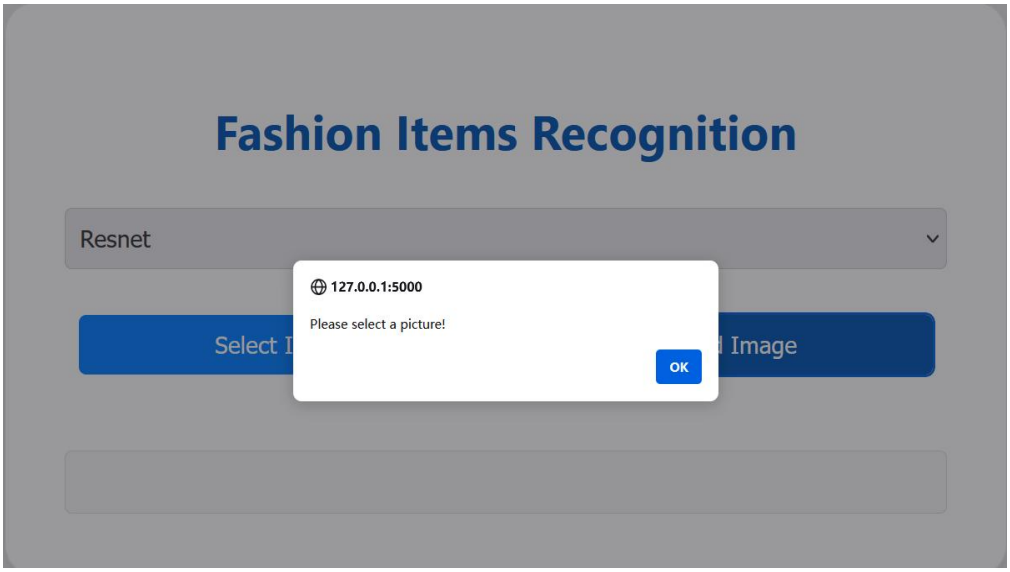

Figure 33 Submit without image

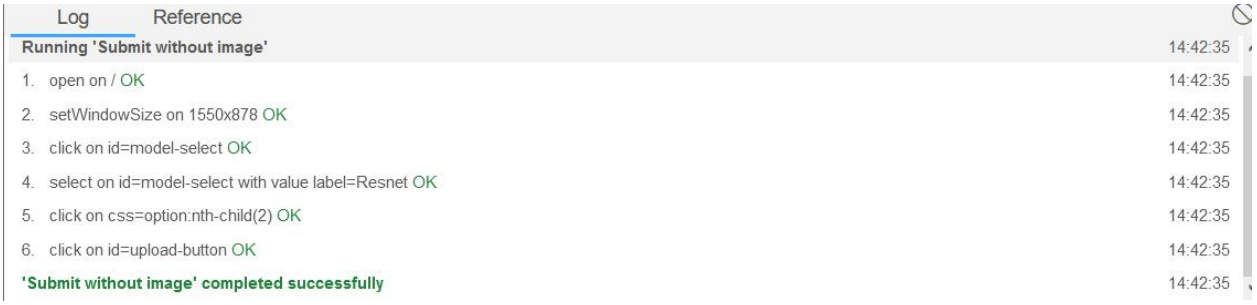

Figure 34 Test log of Submitting without image

C. Submit with model and image

The third test aims to evaluate the scenario where the user selects a model(Alexnet) for fashion item recognition and uploads an image of a backpack. In this case, the web application should proceed to submit the prediction request. Upon completion, it should return the prediction results, including the model name, accurate prediction of the fashion item category(backpack) and the probability scores.

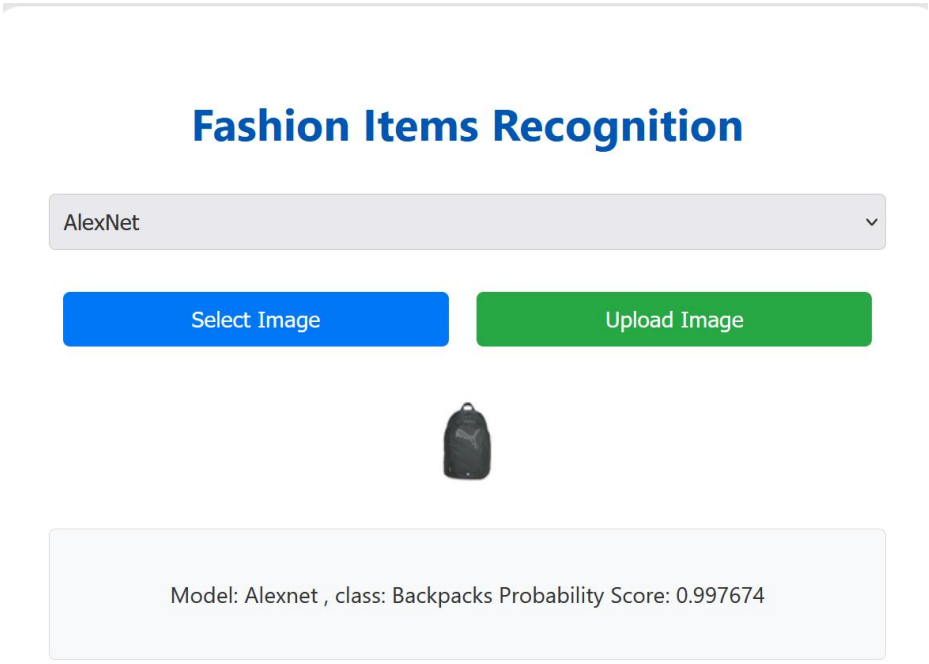

Figure 35 Submit with model and image

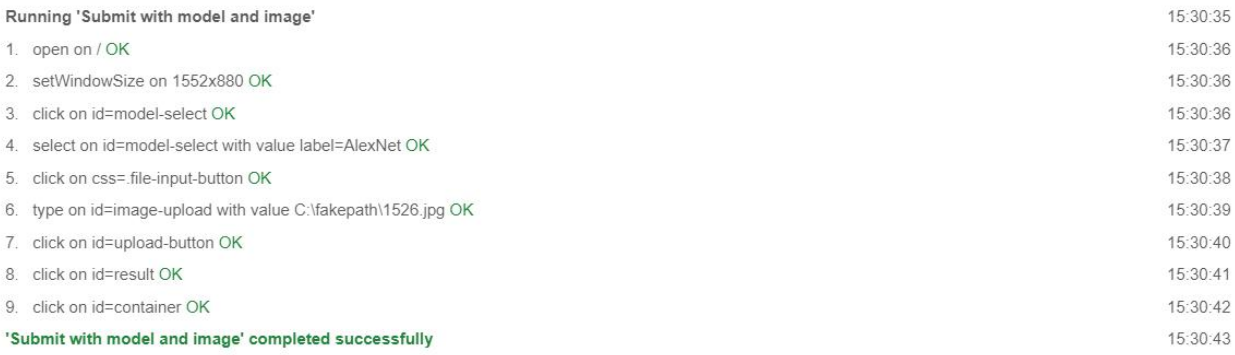

Figure 36 Test logs of submitting with model and image

4.5.2 Manual Testing

Manual testing methods are employed to assess the prediction accuracy of each model for fashion items. Using the same fashion item image for prediction across all models, these tests aim to verify the precision of each model's prediction functionality. Here are the test result in table 17

Test case	Test Data	Expected Results	Actual Results	Status
Use Reset model to recognize a fashion item image correctly	A T-shirts image	The prediction result shows the selected model(ResNet), the predicted class (T-shirt) and the probability score of this prediction	The prediction result shows the selected model(ResNet), the predicted class (T-shirt) and the probability score (0.999959) of this prediction	Pass
Use AlexNet model to recognize a fashion item image correctly	A T-shirts image	The prediction result shows the selected model(AlexNet), the predicted class (T-shirt) and the probability score of this prediction	The prediction result shows the selected model(AlexNet), the predicted class (T-shirt) and the probability score (0.999699) of this prediction .	Pass
Use Convnext model to recognize a fashion item image correctly with the selected.	A T-shirts image	The prediction result shows selected model(Convnext), the predicted class (T-shirt) and the probability score of this prediction	The prediction result shows the selected model(Convnext), the predicted class (T-shirt) and the probability score (0.98953) of this prediction	Pass
Use DenseNet model to recognize a	A T-shirts image	The prediction result shows the selected model(DenseNet), predicted class (T-shirt)	The prediction result shows the selected model(DenseNet),	Pass

fashion item image correctly		and the probability score of this prediction	predicted class (T-shirt) and the probability score of this prediction rate(0.315361) of this prediction	
Use EfficientNet model to recognize a fashion item image correctly	A T-shirts image	The prediction result shows the selected model(EfficientNet), predicted class (T-shirt) and the probability score of this prediction	The prediction result shows the selected model(EfficientNet), predicted class (T-shirt) and the probability score of this prediction rate(0.998002) of this prediction	Pass
Use MobileNet model to recognize a fashion item image correctly	A T-shirts image	The prediction result shows the selected model(MobileNet), predicted class (T-shirt) and the probability score of this prediction	The prediction result shows the selected model(MobileNet), predicted class (T-shirt) and the probability score of this prediction rate(0.595666) of this prediction	Pass
Use ViT model to recognize a fashion item image correctly	A T-shirts image	The prediction result shows the selected model(ViT), predicted class (T-shirt) and the probability score of this prediction	The prediction result shows the selected model(ViT), predicted class (T-shirt) and the probability score of this prediction rate(0.904063) of this prediction	Pass

Table 17 The manual testing results

Chapter 5 Professional Issues

5.1 Project Management

5.1.1 Activities

The table 18 outlines the key objectives and activities undertaken in the project, along with their completion status. It effectively captures the structured approach to developing a deep learning system for fashion item recognition.

Objectives	Activities	Completed
Background research	A1.1 Finding some relevant software and list the specific feature of them A1.2 Reviewing the resources A1.3 Studying the relevant models that would be used in fashion item recognition.	Yes
Dataset Construction	A2.1 Finding the fashion item dataset on kaggle A2.2 Splitting and cleaning the dataset A2.3 Conducting data enrichment	Yes
Model Development	A3.1 Selecting suitable pre-trained deep learning models A3.2 Building seven deep leaning models A3.3 Training and testing each model with suitable hyper parameters A3.4 Export and save the trained models for developing the web	Yes
Supporting web application Development	A4.1 Designing the prototype of the website interface. A4.2 Developing the front-end according to the prototype A4.3 Designing the workflow of front-end	Yes

	and back-end A4.4 Developing the back-end with flask	
Testing and debugging	A5.1 Conducting Functional test to make all the functions meet the requirement A5.3 Designing the test senarios and conduct unit test with test senarios A5.4 Debugging the website and solving the potential problems. A5.5 Recording the test results and making sure all the results are passing	Yes

Table 18 Relevant Activities

5.1.2 Schedule

Here is Gannt chart of the project activities schedule

Figure 37 Gannt chart of project

Here are the details of each activities schedule in table 19:

	Activities	Deadline
1	Project Confirm	2023.10.11-2023.10.13
2	Project Planing	2023.10.14 - 2023.10.18
3	Background Research	2023.10.19 - 2024.10.31
4	Project Proposal Writing	2023.10.25 - 2023.11.03
5	DataSet Building	2023.11.04 - 2023.11.18
6	Models Building	2023.11.15 - 2023.12.14
7	Models Training	2023.12.15 - 2024.01.23
8	Progress Report	2023.12.26 - 2024.01.04
9	Website Front-end Prototype Designing	2024.1.16 - 2024.01.17
10	Model Training Result Evaluations	2024.1.24 - 2024.02.02
11	Website Front-end Developing	2024.02.01 - 2024.02.10
12	Website Back-end Developing	2024.02.06 - 2024.02.17
13	Testing and Debugging	2024.02.18 - 2024.02.27
14	Final Report	2024.02.16 - 2024.04.15
15	Presentation Preparation	2024.04.01 - 2024.05.06

Table 19 project schedule details

5.1.3 Project Data Management

To mitigate the risk of code loss, the code is regularly uploaded to a GitHub repository for backup purposes. This strategy ensures that all changes and progress on the projects are securely stored off-site and can be retrieved at any time. Using GitHub also facilitates version control, allowing for tracking of modifications and easy collaboration. The repository is accessible at the URL: https://github.com/Davidyiiiii/Project

5.1.4 Project Deliverables

To guarantee all the project resources that should be submitted for assessments are clear and precise, they are listed as follows:

- Project Proposal
- Ethics Forms
- Weekly Report
- Progress Report

- Project Code
- Final Report
- Training Logs of each model
- Project Presentation

5.2 Risk Analysis

5.2.1 Risk Identification and Risk Assessment

Risk identification and risk assessment are critical components of project management that help ensure the success and smooth execution of the project. The primary purpose of these processes is to proactively identify potential risks that could impact the project's objectives and assess their likelihood and potential consequences.The matrix in Figure 38 is structured to identify and assess risks associated with the project at technical issues, project management, and organizational challenges.There are four degree of risk which are low, moderate, high and very high.

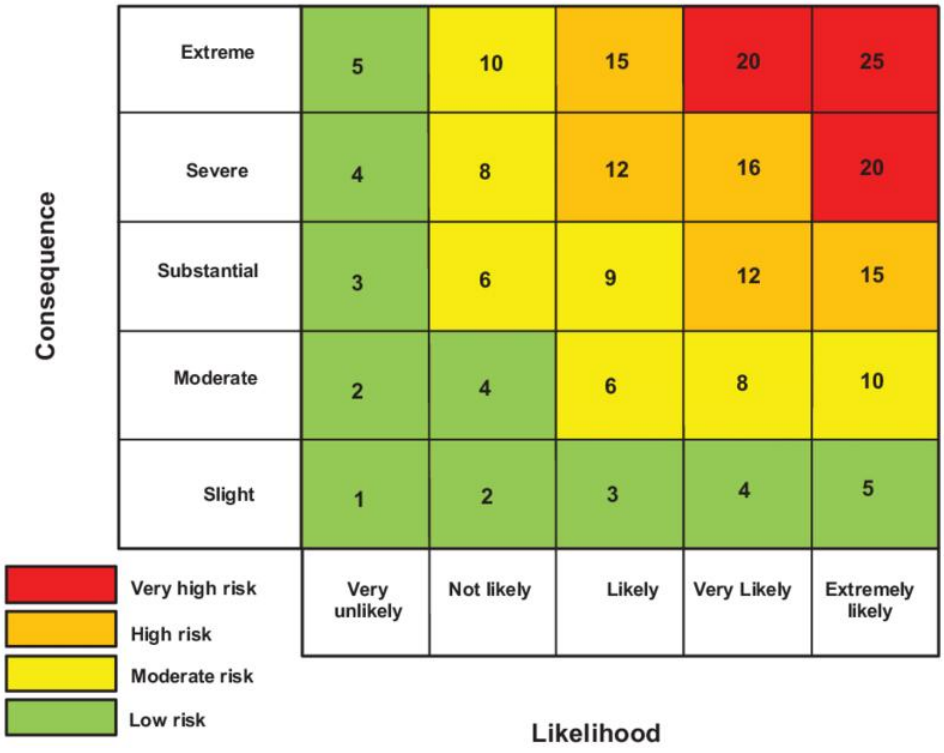

Figure 38 Risk Evaluation Matrix (Dallas,2019)

The identification of risk is in the Table 20 which shows three risk types and the evaluations of the potential risks in this project.

Risk Type	Potential Risk	Causes ID	Consequence	Likelihood	Risk
Technical Issue	Inadequate Dataset Size	C1.1	2	2	4
		C1.2	2	3	6
	Model Overfitting	C2.1	2	2	4
		C2.2	3	3	9
	User Interface Not User-Friendly	C3.1	3	2	6
	Application Bugs	C4.1	3	3	9
		C4.2	4	2	8
	Overlooked Errors During Debugging	C5.1	4	3	12
Project management	Loss of Documentation or Code	C6.1	5	4	20
		C6.2	4	4	16
Organization	Missing Project Deadlines	C7.1	5	2	10
		C7.2	2	2	4

Table 20 Risk identify and risk assessment

5.2.2 Potential Causes and Mitigation Strategy

In order to reduce the influence from the potential risks, it is important to understand the potential causes and how to address the causes. So the table 21 shows the potential causes and the mitigation strategy of each cause, additionally table 20 and table 21 is matched by the Causes ID.

Causes ID	Potential Causes	Mitigation Strategy
C1.1	Limited data sources	Proactively source additional datasets and use data augmentation techniques
C1.2	Limited diversity in dataset	Establish partnerships for data sharing to enhance dataset size and diversity.
C2.1	Insufficient dataset variability	Implement cross-validation
C2.2	Complex models	Hyper-parameter tuning to prevent overfitting
C3.1	Indequate UI/UX design	Perform user testing and Iterate on design based on feedback
C4.1	Inadequate testing	Develop a comprehensive test plan
C4.2	Inadequate quality assurance	Perform automated and manual testing
C5.3	Rushed development cycles	Establish rigorous review, Quality assurance processes and Detailed error logs
C6.1	Inadequate version control	Use version control systems like Git
C6.2	Inadequate backup systems	Regularly backup all project materials to cloud storage solutions and Ensuring recovery options are in place.
C7.1	Poor time management	Implement strict project management protocols
C7.2	Scope creep	Regular progress reviews and contingency

		planning for unexpected delays

Table 21 Potential Causes and Mitigation Strategy

5.3 Professional Issues

5.3.1 Legal Issues

Fashion item recognition using deep learning, addressing legal issues is crucial, particularly concerning data privacy and protection and intellectual property rights. Ensuring compliance with data protection laws, such as the GDPR, involves careful handling of personal data, especially if images might identify individuals. Additionally, respecting intellectual property entails proper licensing and adherence to the terms of use for any datasets and software libraries utilized. Both of these steps are essential to avoid legal liabilities and ensure the project adheres to legal standards.

5.3.2 Ethical Issues

Fashion item recognition using deep learning, ethical issues primarily revolve around ensuring transparency and accountability in how the AI models operate, which includes providing clear documentation and explanations for the system's decisions. Additionally, ethical considerations must address consent and anonymity, especially when dealing with images that might contain identifiable individuals, ensuring that there is proper consent for using these images and that individuals' privacy is respected. This approach upholds the principles of ethical AI usage, focusing on fairness, accountability, and respect for user privacy.

5.3.3 Environmental Issues

The primary concern revolves around the computational efficiency and the resultant carbon footprint of training and deploying AI models. Deep learning models require substantial computational resources which can lead to significant energy consumption. To mitigate the environmental impact, it is essential to focus on optimizing algorithms for greater efficiency and considering the use of energy-efficient hardware. These steps not only reduce the carbon footprint but also contribute to more sustainable technological development.

5.3.4 Professional codes of conduct

Professional codes of conduct, such as those from the British Computer Society (BCS) and the Association for Computing Machinery , serve as ethical guidelines for computing professionals. These codes emphasize the importance of integrity, competence, and responsibility in the practice of computing. They advocate for maintaining high ethical standards, respecting privacy, and prioritizing the public interest in all professional activities. Furthermore, these codes encourage professionals to continuously update their skills, promote fairness and inclusivity, and contribute positively to society while considering the potential social and environmental impacts of their work. Adhering to these codes ensures that professionals not only comply with legal standards but also uphold the trust and confidence of the public in the computing industry.

Chapter 6 Conclusion

6.1 Achievement

This project on fashion item recognition using deep learning stands out for its comprehensive achievements in creating a sophisticated machine learning pipeline tailored for the fashion industry. Starting with dataset construction, a robust dataset was curated from Kaggle, consisting of 43,983 well-defined images spread across 87 categories, providing a strong data foundation. Progressing to model development, seven advanced deep learning models were meticulously selected and trained, including ResNet, AlexNet, ConvNeXt, DenseNet, EfficientNet, MobileNet, and Vision Transformer, each chosen for their unique strengths in image recognition capabilities. A thorough comparative analysis of these models allowed for a nuanced understanding of their performance, examining metrics such as accuracy, loss, precision, recall, and F1-scores, to identify optimal models and techniques for fashion item recognition. The project culminated in the creation of a user-friendly flask-based web application, which operationalizes the deep learning models, allowing users to upload images and obtain predictions on fashion items. This not only demonstrates the practical applicability of the project's outcomes but also provides a gateway for future enhancements and user engagement in the fashion tech space.

6.2 Future Work

A. Dataset Expansion

In future work, dataset expansion will be pivotal, aiming to enrich the model's understanding of the vast diversity found in global fashion trends. This could involve collecting additional images from a wider array of sources, ensuring representation across different cultures, body types, and fashion styles, as well as accounting for variations in lighting, angles, and backgrounds. Integrating user-generated content could provide real-world variability, while collaboration with fashion retailers might offer access to new and evolving clothing lines. Such an expansion would not only bolster the model's accuracy and generalizability but also enhance its inclusivity, ensuring that the system can serve a broader and more diverse user base effectively.

B. Model Enhancements

For model enhancements, future work could focus on exploring state-of-the-art deep learning architectures that push the boundaries of accuracy and efficiency in fashion

item recognition. This includes experimenting with the latest advancements in neural networks, like capsule networks and generative adversarial networks , to handle the intricate details and variety inherent in fashion items. Delving into transfer learning could allow for leveraging pre-trained models on larger datasets to improve performance, especially where data is limited. Additionally, implementing neural architecture search could automate the process of finding the most effective model architectures, optimizing for both precision and computational efficiency. Refining these models could also involve advanced regularization techniques to prevent overfitting and developing ensemble methods that combine the strengths of multiple models to enhance prediction accuracy.

C. Web Application Functionality

In the future, the web application's functionality can be significantly enhanced by incorporating interactive user feedback mechanisms to refine model predictions continually. This can be achieved by allowing users to correct misclassifications, which in turn can be used as new data points for model retraining in an iterative learning process. Integrating real-time fashion trends analytics can personalize and update the recognition capabilities to reflect current styles, potentially through partnerships with fashion databases. Additionally, implementing personalized user profiles can facilitate tailored recommendations, improving user engagement and creating a more individualized experience. These enhancements would not only improve the application's precision and relevance but also heighten the user's interactive experience, fostering a more engaging and dynamic platform.

References

[1] Y.H. Chang and Y.Y. Zhang, 'Deep Learning for Clothing Style Recognition Using YOLOv5', Micromachines, vol. 13, no. 10, p. 1678, Oct. 2022, doi: 10.3390/mi13101678.

[2] W.H. Cheng, S. Song, C.Y. Chen, S. C. Hidayati, and J. Liu, 'Fashion Meets Computer Vision: A Survey'. arXiv, Jan. 28, 2021. Accessed: May 01, 2024. [Online]. Available: http://arxiv.org/abs/2003.13988

[3] L. Alzubaidi et al., 'Review of deep learning: concepts, CNN architectures, challenges, applications, future directions', J Big Data, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[4] S. Bhoir and S. Patil, 'The FASHION Visual Search using Deep Learning Approach'. Sep. 28, 2022. doi: 10.21203/rs.3.rs-2053297/v1.

[5] L. Donati, E. Iotti, G. Mordonini, and A. Prati, 'Fashion Product Classification through Deep Learning and Computer Vision', Applied Sciences, vol. 9, no. 7, p. 1385, Apr. 2019, doi: 10.3390/app9071385.

[6] D. Swain, K. Pandya, J. Sanghvi, and Y. Manchala, 'An Intelligent Fashion Object Classification Using CNN', EAI Endorsed Trans Ind Net Intel Syst, vol. 10, no. 4, p. e2, Nov. 2023, doi: 10.4108/eetinis.v10i4.4315.

[7] S. Shubathra, P. Kalaivaani, and S. Santhoshkumar, 'Clothing Image Recognition Based on Multiple Features Using Deep Neural Networks', in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India: IEEE, Jul. 2020, pp. 166–172. doi: 10.1109/ICESC48915.2020.9155959.

[8] J. Rajalekshmi, S. Nag, R. S. Kharvi, and S. Kumar, 'FASHION APPAREL DETECTION BY MEANS OF DEEP LEARNING TECHNIQUES', JOURNAL OF CRITICAL REVIEWS, vol. 7, no. 04, 2020.

[9] V. Zakaryan, "AI Clothing Detection: Use Cases for Fashion and E-Commerce," Postindustria, 17-Jun-2022. [Online]. Available: https://postindustria.com/ai-clothing-detection-use-cases-for-fashion-and-e-commerce/

[10] Y. Sun, W. K. Wong, and X. Zou, "A Multi-Task Model for Multi-Attribute Fashion Recognition and Retrieval," AATCC Journal of Research, vol. 8, no. 1_suppl, pp. 105-116, 2021. doi: 10.14504/ajr.8.S1.14.

[11] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," SN Computer Science, vol. 2, no. 6, p. 420, 2021. doi: 10.1007/s42979-021-00815-1.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, 2015. doi: https://doi.org/10.1038/nature14539.

[13] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[14] M. Shafiq and Z. Gu, “Deep Residual Learning for Image Recognition: A Survey,” Applied Sciences, vol. 12, no. 18, p. 8972, Sep. 2022, doi: 10.3390/app12188972.

[15] M.S. Ebrahimi and H.K. Abadi, "Study of Residual Networks for Image Recognition," in Intelligent Computing, K. Arai, Ed., vol. 284, Lecture Notes in Networks and Systems, Springer, Cham, 2021, pp. 53, doi: https://doi.org/10.1007/978-3-030-80126-7_53.

[16] A. Pujara, "Concept of AlexNet: Convolutional Neural Network," Analytics Vidhya, 3-Nov-2020. [Online]. Available: https://medium.com/analytics-vidhya/concept-of-alexnet-convolutional-neural-network.

[17] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, ‘A ConvNet for the 2020s’. arXiv, Mar. 02, 2022. Accessed: May 01, 2024. [Online]. Available: http://arxiv.org/abs/2201.03545

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, ‘Densely Connected Convolutional Networks’. arXiv, Jan. 28, 2018. Accessed: May 01, 2024. [Online]. Available: http://arxiv.org/abs/1608.06993

[19] A. G. Howard et al., ‘MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications’. arXiv, Apr. 16, 2017. Accessed: May 01, 2024. [Online]. Available: http://arxiv.org/abs/1704.04861

[20] M. Tan, R. Pang, and Q. V. Le, ‘EfficientDet: Scalable and Efficient Object Detection’. arXiv, Jul. 27, 2020. Accessed: May 01, 2024. [Online]. Available: http://arxiv.org/abs/1911.09070

[21] A. Ghiasi et al., ‘What do Vision Transformers Learn? A Visual Exploration’. arXiv, Dec. 13, 2022. Accessed: May 01, 2024. [Online]. Available: http://arxiv.org/abs/2212.06727

[22] S. Mukherjee, "The Annotated ResNet-50," Towards Data Science, 18-Aug-2022. [Online]. Available: https://towardsdatascience.com/the-annotated-resnet-50.

[23] K. Radhika, K. Devika, T. Aswathi, P. Sreevidya, V. Sowmya, and K. P. Soman, "Performance Analysis of NASNet on Unconstrained Ear Recognition," in Nature Inspired Computing for Data Science, M. Rout, J. Rout, and H. Das, Eds. Cham: Springer, 2020, vol. 871, ch. 3. [Online]. Available: https://doi.org/10.1007/978-3-030-33820-6_3.

[24] G. M. Oktavian and H. Santoso, ‘Leveraging MobileNet, InceptionV3, and CropNet to Classify Cassava Plant Disease’, MIND Journal, vol. 6, no. 2, pp. 183–193, Dec. 2021, doi: 10.26760/mindjournal.v6i2.183-193.

[25] D. Shah, "Vision Transformer: What It Is & How It Works," V7 Labs, Dec.15, 2022. [Online]. Available: https://www.v7labs.com/blog/vision-transformer-guide.

[26] T. Ahmed and N. H. N. Sabab, ‘Classification and understanding of cloud structures via satellite images with EfficientUNet’. Jul. 02, 2021. doi: 10.1002/essoar.10507423.1.

[27] H. Dallas and N. A. Rivers-Moore, Environmental water temperature guidelines for perennial rivers in South Africa. Volume 2: A technical manual for setting water temperature targets. Adaptability and vulnerability of riverine biota to climate change-Developing tools for assessing biological effects View project Shale Gas Development in the Central Karoo: A scientific Assessment of the opportunities and risks View project. [Online]. Available: https://www.researchgate.net/publication/340299110

Appendices

Software Repository：https://github.com/Davidyiiiii/Project