Independent University **Bangladesh (IUB)**

IUB Academic Repository

Computer Science and Engineering | Undergraduate Thesis

2026-04

JewelNet: A Custom CNN and Transfer Learning–Based Approach for Fine-Grained Jewelry Classification

Nishi, Jannatul Ferdous

IUB

https://ar.iub.edu.bd/handle/11348/1166

Downloaded from IUB Academic Repository



JewelNet: A Custom CNN and Transfer Learning–Based Approach for Fine-Grained Jewelry Classification

April 2026

Prepared by:

Jannatul Ferdous Nishi
ID: 2231385

Md. Masum Billa
ID: 2221793

Department of Computer Science and Engineering
Independent University, Bangladesh

Supervised by

Md. Tarek Habib, PhD
Associate Professor
Department of Computer Science and Engineering
Independent University, Bangladesh

Attestation

We hereby declare that this thesis titled "Deep Learning-Based Fine-Grained Jewelry Classification: A Comparative Study of Custom CNN, Transfer Learning, and EfficientNetB2" is our own original work. All materials, ideas, or text taken from published or unpublished works of others have been properly acknowledged and cited following internationally accepted academic standards.
We further declare that no part of this thesis has been submitted elsewhere for any degree or qualification. The work presented in this document is genuine and reflects our independent research effort carried out under the supervision of Md. Tarek Habib, PhD, at the Department of Computer Science and Engineering, Independent University, Bangladesh.

Author Name: Jannatul Ferdous Nishi

Signature: _____

Author Name: Md. Masum Billa

Signature: _____

Letter of Transmittal

April 2026

To
Md. Tarek Habib, PhD
Associate Professor, Department of Computer Science and Engineering
Independent University, Bangladesh

Subject: Submission of Undergraduate Thesis

Dear Sir,
we present our undergraduate thesis entitled "Fine-Grained Jewelry Classification Using Deep Learning: Custom CNN, Transfer Learning, and EfficientNetB2 Comparison" as part of the requirement towards the Bachelor of Science in Computer Science and Engineering degree.
Our study introduces JewelNet, a deep learning-based framework that classifies eight types of jewelry categories by leveraging custom and pre-trained models. It achieves the highest accuracy rate of 95.21% using EfficientNetB2 and expands the capability of the framework to provide jewelry recommendation through deep feature embedding techniques.

It is our utmost pleasure to express our gratitude for your continuous mentoring, supervision, and support during the course of our study. Your invaluable assistance is greatly appreciated.

Thank you very much for your attention.

Sincerely,

Jannatul Ferdous Nishi
(On behalf of the Group)

Evaluation Committee

Supervisor

Name: _____ Signature: _____

Internal Examiner 1

Name: _____ Signature: _____

Internal Examiner 2

Name: _____ Signature: _____

External Examiner

Name: _____ Signature: _____

Acknowledgement

To begin with, first of all, we would like to thank our Almighty God for blessing us with the patience, motivation, and determination to complete this research project. Without His blessings, there wouldn't have been anything for us to accomplish.
We would like to show our utmost gratitude towards our respected Supervisor, Dr. Md. Tarek Habib, Associate Professor, Department of Computer Science and Engineering, Independent University, Bangladesh. His valuable guidance, insightful scholarly feedback, outstanding patience, and immense dedication of his precious time has been instrumental in the successful completion of our thesis. His profound knowledge and experience in deep learning and computer vision have contributed immensely to the quality of our thesis.
We would like to extend our heartfelt gratitude to the Head of the Department of Computer Science and Engineering for creating a conducive academic environment for us to conduct our research successfully. We would also like to thank the Fab Lab IUB for their collaborative space and technical support which has played a significant part in the success of our project.
We sincerely thank our other fellow researchers and peers who gave us constructive feedback and moral support throughout this research process. Their constructive discussions and suggestions have been very helpful to refine our ideas and improve our methodology.
Last but not least, we thank our beloved families for their immense love and constant encouragement and support during this period. Their faith in our capabilities motivated us in each step of this demanding research process.
In conclusion, we would like to acknowledge all those individuals who have directly and indirectly contributed to this research project.

Abstract

Fine-tuned jewelry image classification poses a considerable challenge to the computer vision application due to intraclass similarity, reflective metal material, complex background, and the absence of the corresponding training data. Traditional classification approaches using manually extracted features such as histograms of colors, texture description, and structural attributes have proven ineffective in solving the particular issue due to their limited ability to discriminate between structurally similar classes of items such as bangles and bracelets. Thus, the current research proposes a novel model for solving the given problem named JewelNet, which is an all-inclusive deep learning classifier that utilizes three different architectural approaches to classification including a custom CNN architecture, transfer learning using VGG16 and ResNet50, and the state-of-the-art EfficientNetB2 architecture. Two different sets of experiments were designed based on the same framework to compare various model architectures. A large dataset of 1,217 images of eight different types of jewelry items (bangle, bracelet, chain, earring, necklace, pendant, ring, and nose pin) was assembled. To diversify the collected data, various image augmentation techniques such as rotations, flips, zooming, changes to the brightness, and channels were employed. As a result, the size of the dataset increased to 8,519 images. Experimental evaluation revealed that the EfficientNetB2 architecture achieves the best classification performance with accuracy of 95.21%, with precision and recall being equal to 95.33% and 95.06%, respectively, leading to an F1-score of 94.97%. At the same time, the best classification accuracy for VGG16 is 94.19%, and Custom CNN yields 93.78%. The per-class classification demonstrates that recall for EfficientNetB2 equals 100% for earrings, necklaces, pendants, and rings. Besides, the deep features obtained from this network can be used for recommendation purposes with cosine similarity greater than 0.92.

Keywords: *Fine-grained image classification, jewelry recognition, deep learning, convolutional neural networks, transfer learning, EfficientNetB2, VGG16, ResNet50, data augmentation, content-based recommendation.*

Contents

Attestation **1**

Letter of Transmittal **2**

Acknowledgement **4**

Abstract **5**

1 Introduction **10**
1.1 Background of the Study 10
1.1.1 Problem Statement 11
1.2 Objectives of the Study 12
1.3 Research Contributions 13
1.4 Organization of the Thesis 14

2 Literature Review **15**
2.1 Traditional Image Classification Methods 15
2.2 Deep Learning-Based Approaches 16
2.3 Transfer Learning 17
2.4 Jewelry Classification Using CNNs 18
2.5 Research Gaps 19

3 System Design and Conceptual Architecture **21**
3.1 Proposed Framework: JewelNet 21
3.2 System Workflow 31

4 Research Methodology **34**
4.1 Dataset Collection and Description 34
4.2 Data Preprocessing and Augmentation 36
4.2.1 Image Resizing 36
4.2.2 Normalization 37
4.2.3 Data Augmentation 37
4.3 Custom CNN Architecture 38
4.4 Transfer Learning Models 40
4.4.1 VGG16 40
4.4.2 ResNet50 42
4.5 EfficientNetB2 Model 43
4.6 Training Protocol 46
4.7 Evaluation Metrics 47

5 Experimental Results and Analysis **49**
5.1 Experimental Setup 49
5.2 Overall Performance Comparison 50
5.3 Per-Class Performance: Custom CNN 51
5.4 Per-Class Performance: VGG16 52
5.5 Per-Class Performance: EfficientNetB2 52
5.6 Confusion Matrix Analysis 54
5.7 Training Dynamics 55
5.8 Content-Based Jewelry Recommendation 57

6 Comparative Analysis and Discussion **59**
6.1 Comparison with Existing Works 59
6.1.1 EfficientNetB2 Analysis 60
6.1.2 VGG16 Analysis 60
6.1.3 ResNet50 Analysis 61
6.1.4 Custom CNN Analysis 61
6.2 Practical Implications 61
6.3 Limitations 62

7 Conclusion and Future Works **64**
7.1 Conclusion 64
7.2 Key Contributions 65
7.3 Limitations 65
7.4 Ethical Considerations 66
7.5 Future Work 67

Bibliography **67**

A Dataset Sample Images **73**

B Detailed Hyperparameter Configuration **74**

C Python Code Snippets **75**

D Ethics Approval and Data Declaration **77**

List of Figures

1 Conceptual overview of the proposed JewelNet framework for automated jewelry classification. 21
2 System workflow diagram from input to output. 32
3 Data Processing and Multi-Model Training Workflow 34
4 Custom CNN architecture with layer-wise configuration 38
5 Architecture of the VGG16-based transfer learning model used for jewelry classification. 41
6 Architecture of the ResNet50-based transfer learning model used for jewelry classification. 43
7 EfficientNetB2 architecture with compound scaling 45
8 Confusion matrix showing class-wise prediction performance for the jewelry classification model. 54
9 Confusion Matrix of JewelNet Model 55
10 Training and Validation Performance Curves of the JewelNet Model 56
11 Jewelry Recommendation Output Based on Feature Similarity 57

List of Tables

4.1 Dataset Distribution Before and After Augmentation 36
4.2 Data Augmentation Techniques Applied 38
4.3 Custom CNN Layer-wise Configuration 39
4.4 Training Configuration for All Models 47

5.1 Overall Performance Comparison of All Models 50
5.2 Per-Class Performance Metrics for Custom CNN 51
5.3 Per-Class Performance Metrics for EfficientNetB2 53

6.1 Comparison with Existing Jewelry Classification Methods 59
6.2 Model Computational Efficiency Comparison 62

B.1 Complete Hyperparameter Configuration 74

1 Introduction

1.1 Background of the Study

Jewelry has cultural, personal, and monetary value in many countries around the world. Jewelry made of gold, specifically, indicates status and heritage, as the material has been considered a symbol of status for thousands of years. With digitalization of the jewelry industry in contemporary era, many online marketplaces have emerged. Hence, an urgent need has risen to have smart jewelry image classification systems capable of recognizing various jewelry items based on the images. An obvious need for automation of the process of recognition and classification exists, as there are many practical applications of such technologies. Online marketplace sellers need systems that can classify tens of thousands of jewelry items by labeling them with tags. For retail stores, inventory management systems must be implemented to allow real-time item identification. Finally, visual search engines and recommendation engines need image-matching capabilities in order to make personalized recommendations based on visual similarity [4]. Consequently, it is important to develop advanced algorithms for jewelry classification. Traditionally, jewelry recognition and classification algorithms used classical approaches for feature engineering based on hand-crafting of image features. It was done manually by domain experts and then used along with machine learning classifiers such as SVM and k-NN. Various combinations of color histogram, texture descriptors (such as LBP and Gabor filters), and shape descriptor (like HOG) were used in order to improve the results. Even though such algorithms gave decent results for simple classification problems, they were inefficient for fine-grained classification, as they could not distinguish visually similar classes. However, the limitations of manual feature crafting algorithms become especially evident when dealing with jewelry classification. First of all, rings, bangles, and bracelets have similar circular shapes. Second, chains and necklaces have similar elongated shapes. Moreover, metallic and reflective surfaces of jewelry produce inconsistent appearances at different illumination. Finally, diversity inside of each jewelry category in terms of size, decoration, and designs makes hand-crafted feature templates useless. Introduction of deep learning technologies and CNN architecture has changed the paradigm of image classification. Unlike other algorithms requiring manual crafting of features, CNN models are capable of extracting multi-layered features directly from the raw pixel values. Low-level features, including edges [3], corners, and color gradients, are identified in early convolution layers. With deeper layers, they are merged in higher-level, semantically rich features. Therefore, hierarchy-based approach of feature extraction enables CNN models to find hidden visual features that human experts cannot detect. Another significant breakthrough related to development of image classification techniques is transfer learning. Transfer learning enables utilization of accumulated visual knowledge in pre-trained models with millions of images of 1,000 classes of objects in ImageNet dataset. VGG16, ResNet50, and EfficientNet are among many architectures used for diverse applications from disease recognition to material classification. Clothing classification is another successful application. Yet, there is a lack

of work done in terms of fine-grained jewelry classification [5]. Thus, this thesis attempts to develop a system called JewelNet capable of jewelry classification. Specifically, it examines numerous CNN architectural designs and their performance, including domain-specific custom CNN architecture and transfer learning techniques. Utilizing both approaches enables achievement of highly accurate results.

1.1.1 Problem Statement

In particular, jewelry image classification can be considered as an under-studied computer vision challenge. While conventional image classification relies on the presence of obvious differences between distinct categories in terms of colors, shapes, textures, and other visible characteristics, jewelry classification should be able to deal with categories of objects having very close structural relationships with each other. This is due to the combination of intrinsic properties of jewelry, the environment, and restrictions in previous research approaches. One of the most evident challenges associated with jewelry classification lies in the high level of intra-category diversity and inter-category similarity. First of all, jewelry items belonging to the same category may have various designs, sizes [7], and material compositions. For example, rings may consist of simple metal bands or complicated gemstone-filled constructions, while necklaces may represent objects of many shapes and sizes that may also contain various pendants. Meanwhile, jewelry items belonging to different categories tend to have very close morphologies. Thus, bangles and bracelets can be distinguished by rigidity of their structures while chains and necklaces represent elongated linkages that differ by the presence or absence of a pendant. In this regard, the classifier should possess the ability to differentiate items of visually similar categories, what is rather challenging and should be addressed using deep learning technology. Another difficulty relates to the reflective properties of jewelry materials. Typically, jewelry is produced from metals (gold, silver, platinum) that shine as well as gems that may reflect light. Specular nature of reflections implies that different lighting and perspectives make a huge difference in how items are perceived by a camera. Hence, the same jewelry items produce drastically different patterns of images with bright spots [9], glares, shadows and other distracting artifacts. Such variations reduce the learning capacity of a model to extract invariant features and therefore produce unpredictable results. Moreover, the complexity of background as well as related issues have a paramount importance when it comes to classifying jewelry. The jewellery photos used to be taken against various backgrounds, going from professional studio shots with plain backgrounds through various compositions with one or more objects or images worn by models. This adds to the complexity of classification process, since there would be much distracting information, such as occlusions, for the classification model to consider. Another bias that might emerge due to presence of jewelry in photos with models would be color variations caused by human skin and ornaments worn by people. To be more precise, unlike general-purpose datasets like ImageNet or CIFAR-10, the field lacks any commonly used dataset for fine-grained jewelry classification. Thus, most previous studies use smaller self-made datasets that may not cover all varieties of jewelry items and may be biased in terms of data distribution and data quality. As a result, classification models cannot generalize well and cannot be compared across different pieces of research easily. From the perspective of methods, there also appear to be some issues with jewelry classification in literature [12]. First of all, many studies analyze only one deep learning model without comparing the accuracy of different approaches. This hinders further investigation in which types of neural networks better solve this problem. Second, several studies consider only coarse-grained classification of few jewelry categories, which does not fully describe the task as it appears in practice.

Third, insufficient attention was paid to per-class metrics, which would help to understand in which specific classes a classification algorithm performs poorly. Finally, almost all previous research focuses on solving the classification problem without addressing the question of using the learned representation in other applications [14]. As an illustrative example, there might appear a content-based recommender system or intelligent visual search engine that uses learned representations for similarity calculations. In this context, it can be said that a lack of scalable and well-studied deep learning-based approach towards the jewelry classification problem exists. Therefore, the problem statement here can be said to be the development of an efficient solution for jewelry classification based on deep learning algorithms.

1.2 Objectives of the Study

The primary aim of the proposed research is to develop, implement, and evaluate the effectiveness of the reliable deep learning architecture for accurate jewelry image classification. This study addresses the limitations of the prior research solutions by introducing various architectural innovations, evaluation methods, and applications. The following are particular goals of the proposed research stated explicitly.

- Collection of the jewelry image dataset in real-world settings containing 1,217 images of eight jewelry types with changes in illumination, background color, position of the object, material type, and complexity of its design.
- Proposing a custom CNN architecture designed for fine-grained feature extraction with the use of batch normalization, dropout regularization, and layer optimization techniques.
- Modification of the popular transfer learning models utilizing VGG16 and ResNet50 architectures trained on the ImageNet dataset through systematic freezing of layers and adding classification layers.
- Development of an efficient EfficientNetB2 model based on the concepts of compound scaling and MBConv blocks for improving accuracy-efficiency trade-off.
- Implementation of different data preparation techniques and augmentations for simulating real-world image capture and increasing model generalization ability.
- Comparative evaluation of the models developed in terms of their accuracy, precision, recall, F1 score, false positive rate (FPR), and false negative rate (FNR).
- Evaluation of classification performance in terms of both overall accuracy and each individual jewelry class to identify misclassification patterns and related difficulties.
- Demonstration of the practical applicability of the proposed models for fine-grained classification in developing a jewelry recommendation system.
- Establishing a reproducible benchmark for fine-grained classification of jewelry images with recommendations on future research in the area.

1.3 Research Contributions

This thesis offers a number of new and unique contributions to the field of fine-grained image classification. In particular, contributions of this research to domain-specific fine-grained image classification can be summarized as follows:

- **JewelNet Framework:** JewelNet is proposed by this study as an end-to-end deep learning model for jewelry classification task. Several different types of architectural approaches are used within this model including custom architecture CNN, transfer learning based VGG16 and ResNet50 models, and compound scaled architecture EfficientNetB2. This modular approach allows the use of multiple neural networks under consistent experimental settings and facilitates the comparison of different types of architectures on the same test set.

- **Eight-Category Jewelry Dataset:** In this research, a comprehensive, real-world dataset consisting of 1,217 high-resolution images of jewelry belonging to eight categories was compiled and used in experimentation. In addition, the dataset includes substantial variations of lighting conditions, background types, orientations, materials, and jewelry designs. Additionally, the dataset underwent extensive data augmentation that resulted in its expansion to 8,519 images. This dataset addresses one of the issues associated with jewelry classification, namely the lack of domain-specific databases.

- **High Accuracy Fine-Grained Classification:** The EfficientNetB2-based network architecture was found to exhibit high classification accuracy and outstanding values of other metrics, such as precision, recall, and F1-score. These results show the effectiveness of compound-scaled architectures in capturing minor visual differences in the image. Moreover, the results indicate that compound-scaled architectures (represented here by EfficientNetB2) provide a greater ability to extract informative features from the input image compared to traditional convolutional neural networks and transfer learning methods.

- **Multi-Model Comparison and Comprehensive Evaluation:** In this study, a comparison between four deep learning architectures was conducted using the same experiments. The following architectures, including Custom CNN, VGG16, ResNet50, and EfficientNetB2 have been used. To ensure a comprehensive evaluation, multiple metrics, including accuracy, precision, recall, F1-score, false positive rate/false negative rate, and confusion matrix have been employed.

- **Per-Class Analysis and Error Investigation:** Alongside with evaluating global metrics, class-wise performance evaluation has been undertaken in this study as well. In particular, error analysis of classes with similar structure features has been examined in this study, including bangles/bracelets and chains/necklaces. Such class-wise investigation may help in further improving the architecture of the neural network.

- **Content-Based Jewelry Recommendation System:** Apart from the task of fine-grained classification, this study examines the possibility of utilizing deep feature embeddings for practical tasks, such as a content-based recommendation system. Deep feature embeddings, obtained through EfficientNetB2 architecture, are used to estimate similarities between jewelry based on visual similarity using cosine similarity. This system allows establishing similarities scores exceeding 0.92 for similar items.

- **Computationally Efficient Architecture for Practical Deployment:** Alongside fine-grained classification task, all evaluated architecture performance metrics were compared to identify an efficient architecture in terms of computational efficiency. Such metrics include model size, training time, inference speed. EfficientNetB2 proved to provide efficient trade-offs between these metrics, making it suitable for practical applications.
- **Establishment of Experimental Benchmark for Jewelry Classification:** This research provides the basis for experimental benchmarking of fine-grained jewelry classification by introducing a test dataset, processing methods, and evaluating procedure.

1.4 Organization of the Thesis

The following six chapters will address the remaining topics in this study. These chapters will be presented in an organized and logical manner, in order to ensure coherence throughout.

Chapter 2 will present an extensive literature review. This chapter covers traditional approaches in image classification which involve feature extraction and classic machine learning methods. It will discuss the development of deep learning methods, including various convolutional neural network architectures and efficient training approaches such as transfer learning. It will also introduce state-of-the-art CNN models like EfficientNet. Lastly, past studies on jewelry classification will be introduced together with some research gaps which motivate this study.

Chapter 3 will cover the design and architecture of JewelNet. In this chapter, the architecture of the entire system will be covered in details including the workflow from the input images to the classification output. Specifically, this chapter will cover data acquisition, preprocessing, modeling, model training, validation and evaluation, as well as recommendations. Important aspects such as computation efficiency will also be addressed.

Chapter 4 will cover the entire research methodology. It will go over dataset collection and data preparation procedures. It will describe the data augmentation techniques to improve the model generalization ability. This chapter will also elaborate on how the custom CNN was designed and trained. It will describe how transfer learning models and EfficientNet were adapted and fine-tuned in this research. Lastly, it will explain the training protocol and evaluation process.

Chapter 5 will discuss the performance of the implemented models in great detail. This chapter will compare the overall performance of different models using various metrics like accuracy, precision, recall, F1 score, etc. Per-class performance will be discussed and evaluated. Classification results will also be discussed with emphasis put on error analysis of visually similar classes. Moreover, training and validation dynamics will be covered. The functionality of the content-based jewelry recommendation system will also be demonstrated.

Chapter 6 will conduct a critical discussion of the findings and implications of this study. It will first compare this approach with existing works in the literature. It will highlight the pros and cons of different CNN architectures implemented in this study. Practical applications and limitations will also be covered in this chapter. Challenges and difficulties met during this study will be discussed as well.

Finally, the conclusions of this thesis will be drawn in Chapter 7. The overall efficiency of the JewelNet solution will be evaluated. Some ethical considerations will be brought up. Possible future research directions will be presented.

2 Literature Review

2.1 Traditional Image Classification Methods

Automated image classification has seen great advancements over the past decades, moving from rule-based systems to modern, data-driven methods. Early image classification was characterized by the heavy utilization of handcrafted feature descriptors, in which explicit domain knowledge was incorporated into the design of feature descriptors [16]. Such approaches constituted the backbone of classical computer vision before deep learning became mainstream [38, 16].

Handcrafted feature-based approaches are commonly grouped into three main types, namely color, texture, and shape descriptors. In general, each descriptor type tries to capture the respective feature of the image and thus enables traditional classification based on machine learning algorithms [17].

Among the most prominent and popular early approaches are color-based descriptors. Particularly, color histograms provided a statistical representation of the colors contained in an image. While computationally efficient [19], such descriptors are highly prone to changes in the lighting conditions and ignore important spatial information, limiting their applicability. Color correlograms [23], a variant of color histogram descriptors, attempt to take into account such spatial information, yet proved to be inefficient in diverse environments.

Another approach to improve on previous work has been texture-based features. Local Binary Patterns are an approach for capturing microtexture information using a comparison of the intensities of the neighboring pixels and therefore being highly resistant to monotonic changes in lighting [27]. Similarly, Gabor filters mimic human vision and provide orientation-invariant texture features. These are especially useful in analyzing the surface and patterns of jewelry articles [40, 72].

In turn, shape-based descriptors focus on extracting structural characteristics of an object present in an image. Popular approaches include Histogram of Oriented Gradients, which extracts robust keypoints; Scale-Invariant Feature Transform and Speeded-Up Robust Features that capture both keypoints and edge information in order to provide a more holistic view of an object [27]. Such features have been successfully used for representing an image as a bag-of-visual words model.

Typically, handcrafted feature representations are coupled with traditional machine learning classifiers such as SVMs, random forests, or k-Nearest Neighbors. For example, Usha and Perumal [68] achieved successful results in content-based image retrieval through a combination of color and texture features coupled with SVM classification. Early classification systems in other areas have used similar approaches in achieving reasonable performance on structured datasets [31].

However, despite being widely used and generally successful, traditional image classification approaches are not without flaws [32]. Firstly, they face an acute problem known as *the semantic gap*, meaning the inability of low-level pixel information to effectively represent

object semantics. Specifically, while handcrafted features can capture basic visual properties of an image, they are ineffective for capturing object semantics required for accurate fine-grained classifications [10, 15].
Furthermore, traditional handcrafted image processing methods require significant domain knowledge and manual parameter tuning, making them hard to apply to new datasets and problems [35]. They also tend to be extremely fragile to variations in environment conditions such as light, angle, occlusions, or background clutter [6, 55]. Indeed, in jewelry images, surface reflection may alter the appearance of color and texture, rendering these features less reliable.
Moreover, handcrafted approaches become inefficient as image datasets become larger and richer in structure [36], making scaling a crucial issue. In addition, these methods are unable to create hierarchical features for more complicated visual information analysis.
Research carried out in related domains has revealed additional weaknesses of traditional image classification methods. For example, in medical image classification [35] and remote sensing [6], researchers found that traditional methods struggled to achieve high performance compared to deep learning approaches. In object recognition and detection systems, similarly limited results were observed [49].
Ultimately, these factors motivated the move to deep learning-based approaches, which eliminate the need for manual feature extraction and enable the creation of hierarchical representations directly from data. The challenges faced by traditional methods [39], especially in fine-grained tasks like jewelry classification, highlight the need for more advanced and adaptive techniques.

2.2 Deep Learning-Based Approaches

The advent of deep learning revolutionized the realm of image classification by providing means for the automatic learning of hierarchical feature representation from the raw image pixels [41]. As opposed to previous feature-based techniques, deep learning approaches no longer involve handcrafted features but learn discriminative features via an optimization procedure. The main idea behind deep learning is that a series of convolutional, activation and pooling layers transform the input image into higher level of abstraction [38, 16]. Such hierarchy of features helps to solve the problem of semantic gap that was previously hard to overcome [55, 26].
One of the most important breakthroughs of deep learning based image classification was the design of AlexNet architecture by Krizhevsky et al. [37]. This architecture outperformed classical feature-based models by showing remarkable accuracy when processing the ImageNet dataset [16].
A number of architectures were subsequently introduced aiming at further performance improvements and better scalability. For example, the VGGNet architecture [56, 57] showed that an increase in the network depth using small filters (3×3) improves the performance. However, deep networks suffer from vanishing gradient problem and are costly to compute.
To mitigate such drawbacks, ResNet [24] introduced the concept of residual learning using skip connections, helping to propagate gradients between layers [42]. Such design allowed to train extremely deep networks without compromising the performance. Since then, a number of ResNet-based architectures have been successfully applied in various domains [34, 49].
Inception architectures [63] represent yet another breakthrough allowing to extract multi-scale information at the same layer of the network by performing convolution operations

using kernels of various sizes simultaneously [47].
Some of the most recent works focus on improving efficiency and utilizing attention mechanisms. The work of Tan et al. [65, 66] introduced the idea of compound scaling by simultaneously scaling the network depth, width, and input resolution. The attention mechanism was introduced into deep learning by Hu et al. [28] allowing networks to pay more attention to informative channels.
Transformer architecture has recently been adapted for image classification tasks resulting in the development of architectures like Vision Transformers (ViT) [18] and Swin Transformers [43]. These architectures make use of self-attention mechanism which helps them capture long-range image dependencies. However, ViT is computationally expensive, thus making it suitable only for larger datasets.
Deep learning models show outstanding performance in fine-grained classification tasks where subtle differences between visually similar classes should be identified [10, 15].
Deep learning approaches bring numerous advantages to jewelry classification. Jewelry images are usually characterized by complex textures, reflective surfaces[50], and intricate designs which are very hard to model in hand-crafted features. CNN-based models provide a solution to this problem by being able to learn those patterns directly from the image [40, 72, 74].
There are some examples of applying deep learning to jewelry classification [59, 70, 4, 3]. For instance, Singh and Kaewprapha [59] showed that CNNs produce better accuracy. Vaibhav et al. [70] stressed the importance of the diversity of training dataset. Alcalde-Llergo et al. [4, 3] proposed encoder-decoder architecture obtaining state-of-the-art results.
Deep learning approaches are also successfully applied in domains closely related to jewelry such as gemstone classification [19, 29] and materials analysis [48].
However, there are several limitations of deep learning approaches to jewelry classification. Firstly, a huge amount of labeled data is required to train deep networks. Secondly, deep models are computationally expensive which requires additional computational resources. To cope with these disadvantages, the data augmentation technique [51] and the transfer learning method [76, 35, 7] are typically utilized.
Furthermore, interpretability of deep learning models is rather poor.

2.3 Transfer Learning

Transfer learning has gained popularity as one of the most important paradigms in modern deep learning approaches. It relies on transferring the knowledge acquired by a deep neural network through training on a large amount of labeled data in one domain into another [21], related domain. Such training reduces the amount of necessary labeled data and saves resources. In computer vision applications, transfer learning is generally achieved via models trained on huge annotated datasets, such as ImageNet [16].
The rationale for transfer learning is based on the hierarchy of knowledge acquired by CNNs at different layers [58]. Features learned by early layers in such neural networks are rather general and include such basic things as edges [52], corners, color gradients, and other visual elements, regardless of the specific domain of application. Deeper layers are responsible for encoding more specific information about objects in images [38].
Several approaches may be applied to implement transfer learning, depending on how similar the source and target datasets are. Common approaches include feature extraction [61], when the entire network is frozen as a feature extractor, and fine-tuning, when part or all of the network undergoes additional training on the target dataset.
Zhuang et al. [76] present a detailed review of transfer learning paradigms and demonstrate

that they allow improving model accuracy significantly, especially when small datasets are used. Similar conclusions were reached by Kim et al. [35], who investigated the problem in the field of medical imaging, and Bhoir and Patil in the case of e-commerce image classification.
Transfer learning allows achieving good performance in fine-grained classification problems [53], where discrimination between visually similar classes requires very discriminative features [15, 10].
In addition to increasing accuracy, transfer learning improves convergence rates and decreases training times, helps avoid overfitting, and allows working with deep networks, such as VGG16 [56], ResNet50 [24], or EfficientNet [65].
Transfer learning has proven itself useful in such areas as medical imaging [35, 55], remote sensing [6], and industry-related applications, such as product recognition [7, 33].
For fine-grained classification in a domain like jewelry image classification, where labeled data is scarce [44], images contain complicated textures and vary in terms of illumination, transfer learning can be highly beneficial due to pre-training.
Several works have used this methodology, including Singh and Kaewprapha [59] and Vaibhav et al. [70] in the case of jewelry classification, as well as Alcalde-Llergo et al. [4, 3], who achieved good accuracy in this problem. In addition, in related problems, such as gemstone classification [19, 29] or jewelry material assessment [48], transfer learning proved effective.
Still, this methodology faces certain difficulties [54], including domain mismatches and the possibility of overfitting or catastrophic forgetting during fine-tuning.
Overall, transfer learning can significantly improve accuracy and reduce training time in fine-grained image classification problems.

2.4 Jewelry Classification Using CNNs

Jewelry image classification belongs to fine-grained recognition sub-domains, where the goal is to discriminate between categories of objects that are visually very similar and differ in small structural details [64]. This problem is not as popular as such applications as object detection, medical imaging, and facial recognition, as it is rather difficult to obtain appropriate datasets and to ensure good classification accuracy due to complicated textures [11], reflective material properties, and very high inter-class similarities [65].
One of the pioneering works in the field of jewelry classification using CNNs was presented by Singh and Kaewprapha [60]. They compared traditional approaches based on feature extraction using AlexNet [37] followed by SVM classification with deep learning methods based on Inception architecture [63]. Results have shown that deep learning yielded higher classification accuracies within the range of 88
Another work on CNN-based jewelry classification was performed by Vaibhav et al. [71]. They also managed to achieve accuracies near 90% using relatively shallow networks; however, the dataset included only images of Indian jewelry.
More advanced deep learning approaches were employed by Alcalde-Llergo et al. [1] who introduced a hybrid deep architecture based on a combination of VGG16-based feature extraction [56] and GRU recurrent units for recognition and description. This allowed them to achieve about 93% accuracy, which was slightly improved in their next paper to 94% [2].
Some works related to jewelry image classification were carried out in closely related domains, such as Freire et al. [20], who developed an approach to gemstone classification using transfer learning and achieved about 84% accuracy. In addition, Huang and Cui [30]

introduced a novel architecture called MCNN+ using multi-feature fusion for similar purposes, while Meng et al. [46] addressed the issue of assessing the quality of jewelry materials.
Some recent works focused not only on classification but also on recommendation systems based on visual similarity search, as in the case of Islam et al. [33], who discussed visual similarity-based retrieval in fashion applications. At the same time, Sulthana et al. [62] were able to improve recommendation accuracy in e-commerce applications using CNN-based feature extraction.
However, there still are some challenges, such as high inter-class similarity between, e.g., bangles and bracelets [73], changes in lighting conditions, reflective effects, occlusions, and other factors. Thus, the models should be rather accurate and effective, and proper preprocessing is required.
Recent trends involve deeper architectures, transfer learning, and feature extraction. Some examples can be found in such papers as Cui et al. [15], Chengcheng et al. [10], and Yang [74].
Overall, the use of deep learning and specifically CNNs significantly improves the accuracy in jewelry classification. However, there is a lack of works devoted to this problem.

2.5 Research Gaps

A review of existing literature highlights a number of important gaps that prompt us to carry out this research [69]. To begin with, the vast majority of previous studies use relatively small databases, comprising no more than 1,000 images. The limited size of databases constrains deep learning algorithms' generalization capability and their performance on real-life tasks. Although data augmentation can mitigate this gap to some extent [51], this technique cannot substitute the actual variety of the data.
Secondly, many of the previously performed studies target a few jewelry categories (three to five classes). A much greater variety of jewelry items should be considered in order to build more robust models and make them applicable in practice. Moreover [75], jewelry databases include not only classes but also sub-classes, i.e., more complicated structure of classes.
Thirdly, there is insufficient analysis of different deep learning architectures. Most studies test only one model, thus being unable to assess the pros and cons of using VGG, ResNet, EfficientNet, etc. Comparative studies (Mascarenhas et al. [45]) stress the necessity to analyze multiple architectures under identical conditions.
Fourthly, evaluation procedures used in existing studies are mostly based on overall accuracy. While this metric provides the idea about the quality of the classifier in general terms [25], it does not take into account such factors as class imbalance, false positives/negatives, etc. Comprehensive evaluation procedures include metrics such as precision, recall, F1-score, and confusion matrix analysis.
Fifthly, most researchers regard jewelry classification as a standalone problem and do not integrate it into any practical application (recommendation system, visual search engine, etc.). However, modern e-commerce systems require integration of AI components and their usage in combination with each other [13]. Therefore, it is important to create a solution that can be embedded into larger systems.
Sixthly, advanced architectures, including EfficientNet [65], Vision Transformer [18], and their combinations, have never been tested in jewelry classification. These architectures provide better results in generic image classification and could potentially be used to improve fine-grained recognition performance.

Lastly, the specific nature of jewelry classification tasks implies the need for additional consideration of certain features [8]. Specifically, visual similarity between classes (such as bracelet and bangle) remains one of the problems that still require solving in order to improve model performance. Advanced techniques for feature extraction [22], in particular, attention mechanism and part-based modeling, can help solve this problem.
This thesis aims to fill all the described gaps and introduce a modular framework that would allow testing multiple deep learning architectures (EfficientNet, VGG16, ResNet50, as well as a custom CNN architecture) with the help of an elaborate procedure and diverse data [67]. The framework goes beyond classification and implements content-based recommendations, thus enabling integration into practical systems.

3 System Design and Conceptual Architecture

3.1 Proposed Framework: JewelNet

JewelNet is an innovative approach to deep learning which proposes an end-to-end pipeline for handling fine-grained jewelry image classification. The very essence of jewelry poses numerous challenges to any image recognition algorithm because it implies a high level of intra-class variance and inter-class similarity in addition to reflective properties of jewelry and intricate backgrounds.
JewelNet can be regarded as a learning framework that combines a variety of state-of-the-art deep learning paradigms in one experimental setup. In contrast to typical deep learning systems that utilize only one neural network architecture, JewelNet integrates several popular networks, including domain-specific Custom CNN, transfer learning models (VGG16 and ResNet50), and a more contemporary EfficientNetB2 model. Combining models of various types, JewelNet is capable of leveraging generalizable features of transfer learning models, while at the same time being able to detect subtle differences between categories thanks to domain-specific customized architectures.
Besides standard classification of images, JewelNet provides extra functionality in form of similarity-based recommendation. Such additional functionality allows not only determining the class of a given jewelry sample but also performing similarity search based on deep feature vectors of the samples. Thus, JewelNet may be considered a tool for practical applications in e-commerce sites, intelligent recommendation engines, product catalogs, and many other areas.

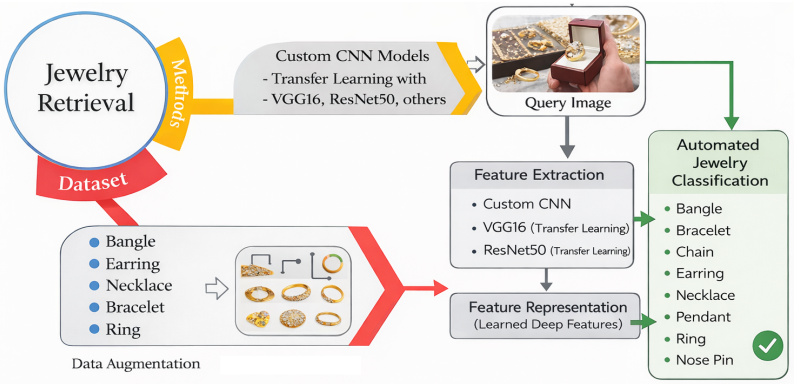


Figure 1: Conceptual overview of the proposed JewelNet framework for automated jewelry classification.

As illustrated in Fig. 1, demonstrates a conceptual view of the JewelNet framework. It includes several key steps of data processing and learning pipeline that transform an input image into output class and feature embedding vectors.

At the system-level abstraction, JewelNet can be described by a mathematical expression that denotes a mapping function of the following form:

$$\mathcal{F} : I \rightarrow (\hat{y}, \mathbf{f})$$

where I represents an input image; $\hat{y}$ is the predicted class label; and $\mathbf{f}$ is a feature vector coordinate. The equation given above shows how the network is able to simultaneously classify and represent images as features.
The creation of the JewelNet is based on three concepts that guarantee the framework's robustness, flexibility, and applicability in practice.

- **Modularity:** All steps of the JewelNet framework are implemented separately. This feature ensures that components of the framework could easily be replaced by some other architectures, techniques, or algorithms without changing the whole framework. ‘‘‘
- **Reproducibility:** All aspects of experiments were carefully controlled and documented. This allows the results produced by the framework to be reproduced under any circumstances.
- **Practical Applicability:** The JewelNet framework can be used in real life since it solves two critical problems – classification and feature extraction for image comparison.

In addition to these concepts, efficient computations and scalability were among the objectives during the creation of the JewelNet. For example, the use of EfficientNetB2 in the framework helped to achieve the required trade-off between classification and computational performance thanks to compound scaling and efficient convolutional layers. JewelNet is consisted of six key modules that form a sequential data processing pipeline to solve a fine-grained jewelry classification problem.

1) Data Acquisition Module

The Data Acquisition Module is responsible for the systematic acquisition, classification, and validation of the dataset used to train, test, and validate the JewelNet model. Because the accuracy of any deep learning algorithm depends on the training data set, the Data Acquisition Module plays an essential role in the proposed solution.
The dataset is compiled from different heterogeneous resources, such as real-life photography, ecommerce sites, and even official catalogs. Multi-resource data acquisition ensures that the dataset contains a wide range of images with different lighting, background complexities, orientations, and looks – which is necessary in order to train models that will be able to work successfully outside of the lab environment.
The dataset consists of eight fine-grained jewelry categories:

$$\mathcal{C} = \{\text{bangle}, \text{bracelet}, \text{chain}, \text{earring}, \text{necklace}, \text{pendant}, \text{ring}, \text{nose pin}\}$$

Every class is characterized by its own unique but possibly overlapping category, making the task of classification quite complicated. Special focus is placed on the classes having highly similar structures, like chains and necklaces or bangles and bracelets, requiring careful labeling.
In the first test, the researchers employed a well-balanced set containing 1600 images (200 images per class) to make sure that all the classes were equally represented during the

training process . For the second test, an unbalanced dataset of 1217 images taken from reality was assembled to better reflect the natural differences between the classes .
To assure the high quality of the created dataset, the following conditions should be satisfied for every single image:

- Only one main item of jewelry appears in each image to maintain consistent labeling.
- Images that have heavy occlusion, high noise, and unclear class labels are omitted from the dataset.
- Near duplicate images are deleted to avoid possible data leakage.

After preprocessing, the dataset is sorted into folders in accordance with its class labels. Let D be the dataset:

$$D = \{(I_i, y_i)\}_{i=1}^{N}$$

Where I_i is the i^{th} image and $y_i \in \mathcal{C}$ is the associated label.
The dataset was divided into three disjoint sets to facilitate supervised learning:

- Training set (70%) for model learning
- Validation set (15%) for hyperparameter tuning
- Test set (15%) for final evaluation

In this way, an estimate can be obtained regarding the generalizing capabilities of the model by assessing the performance of the model on unseen data.
Moreover, the design of the dataset includes elements of jewelery images which make them more complicated to process, namely reflection, metallic qualities, gemstones, and uniqueness of each image. All these aspects of images make it necessary to have strong feature learning algorithms in later parts of the JewelNet architecture.
Finally, it can be concluded that the Data Acquisition module provides an important and highly varied dataset required to train and test the JewelNet system efficiently.

2) Preprocessing and Augmentation Module

Preprocessing and Data Augmentation Module deals with the conversion of the initial input images into an enhanced form that can serve as an input to the deep learning algorithm. This step is very important, since it ensures numerical stability, faster convergence during training, and increased generalization capability of the model.
Considering the different image resolutions, illumination conditions, complex backgrounds, and placement angles of the objects within the images, a structured procedure for data preprocessing is carried out. Each image $I \in R^{H \times W \times C}$ is scaled to ensure spatial consistency in the resolution of all images. In case of using neural networks such as VGG16 and ResNet50, the size of images should be:

$$I' \in R^{224 \times 224 \times 3}$$

As far as EfficientNetB2 is concerned, the application of marginally higher resolution (e.g., 260×260) can be taken into account as meeting the requirements of the architecture of the network.
Following rescaling, the values of pixels are normalized in the range of $[0, 1]$:

$$I'' = \frac{I'}{255}$$

This step helps reduce the scale of input values, stabilize gradient updates, and speed up the convergence of the training process. In some cases, additional normalization techniques, such as mean subtraction or standardization, may also be applied to further improve model performance.

To handle the limited size of the dataset and avoid overfitting, a rigorous **data augmentation technique** is used. Data augmentation increases the variability of the training dataset by performing random transformations on the input images. This can be mathematically represented as:

$$I_{aug} = \mathcal{A}(I'')$$

Where $\mathcal{A}$ is defined as a collection of stochastic transformation operators used during the training phase.

The various augmentation procedures used in this technique framework are as follows:

- **Rotation:** Arbitrary rotation of the image within a certain range (e.g., $\pm 20°$).
- **Translation:** Translating the object horizontally and vertically to deal with displacements within the scene.
- **Scaling and Zooming:** Arbitrary scaling and zooming of the object to simulate variation in sizes.
- **Shear Transformation:** To achieve robustness against perspective distortion.
- **Horizontal Flipping:** Reflection of images to introduce more variety in the data set.
- **Brightness Adjustment:** Changes in light intensity to mimic various lighting situations.
- **Channel Shifting:** Small changes in color channels to address variations in materials.

Indeed, the usage of transformations is randomly chosen at each step in the process of learning, which means that the model is provided with input images that have been changed in one way or another. This significantly increases the ability of the model to generalize about the data it has not seen before, as in the case of real-life application, jewelry items might be seen in all sorts of positions and under various lighting and environmental conditions.

Moreover, data augmentation helps solve the issue of class imbalance and the small size of the dataset. It is also crucial because it prevents the model from merely memorizing the training dataset patterns.

Speaking of domain specifics, it should be noted that the reflective surface of metal parts of the jewelry and the detailed construction of jewelry pieces require taking into account their appearance at different angles of vision and under various illumination conditions.

Thus, it is safe to conclude that the Preprocessing and Augmentation Module plays its part in preparing the input dataset for further processing.

3) Model Zoo

Model Zoo can be regarded as the most critical component of JewelNet as it includes all deep neural network architectures that are employed within this work. The reasoning for the employment of multiple architectures is based on the ability to evaluate the performance of each of them under a different approach to the design of the architecture itself, such as custom design, traditional transfer learning, and efficient architecture design.
Each network included into the Model Zoo is fed with an input image I_{aug}, which results in obtaining a feature vector $\mathbf{f}$.

$$\mathbf{f} = \phi(I_{aug}; \theta)$$

where ϕ denotes the feature extraction function and θ represents the model parameters.
The Model Zoo includes four distinct architectures, each representing a different design philosophy:

a) Custom CNN Architecture

The Custom CNN is purposely designed to extract specific features in jewelry images. The model architecture involves several convolution blocks where the number of filters gradually increases with each block, such as 32, 64, 128, and 256.
Each convolutional layer performs the following operation:

$$(I * K)(x, y) = \sum_m \sum_n I(x - m, y - n)K(m, n)$$

where K denotes the convolutional kernel. Batch normalization, ReLU activation function, and max-pooling layer follow the convolutional layers to add stability, non-linearity, and dimensionality reduction respectively.
To avoid overfitting caused by the small data set, dropout layers and L2 regularization have been added to the model. Full connection layers apply classification using a Softmax activation function.
Custom CNN is especially useful for recognizing local patterns like those found on gemstones, metals, and decorations.

b) VGG16 Transfer Learning Model

The VGG16 model is a deep convolutional neural network that contains 16 layers with weighted values. It uses small 3×3 filters for each convolution layer.
Within this model, the VGG16 neural network will be pretrained using ImageNet. Lower layers will be frozen to keep general feature extraction capabilities, whereas higher layers will be tuned to perform better on jewelry.
Instead of the classification layers, custom layers will be used:

$$\mathbf{f} \rightarrow \text{Dense}(512) \rightarrow \text{Dense}(256) \rightarrow \text{Softmax}(8)$$

VGG16 shows excellent performance because of its capability to extract features at both lower and higher levels, giving outstanding accuracy in fine-grained classification.

c) ResNet50 Architecture

The ResNet50 model uses residual learning with skip connections to train deep models without suffering from degraded performance. The key concept of residual learning can be

stated as follows:

$$y = F(x) + x$$

where x denotes the input, and F(x) is the residual function learned by the neural network. The above architecture facilitates easier computation of gradients during backpropagation, hence preventing the vanishing gradient issue. ResNet50 comprises multiple residual blocks in a hierarchical form, facilitating deep learning.
For the purposes of this research, ResNet50 will be fine-tuned by keeping some of the layers fixed while training the deeper layers of the network. While the network performs well in terms of feature representation, it is also sensitive to training data.

d) EfficientNetB2 Architecture

EfficientNetB2 is an advanced network architecture that offers excellent performance with lower computational complexity through compound scaling. In contrast to conventional networks that independently increase depth, width, or resolution, EfficientNet increases all three dimensions at once:

$$\text{depth} = \alpha^{\phi}, \quad \text{width} = \beta^{\phi}, \quad \text{resolution} = \gamma^{\phi}$$

where α, β, γ are scaling factors and ϕ is the combined scaling factor.
The network architecture uses MBConv and SE layers, which allow for effective feature extraction and channel-level attention.
EfficientNetB2 is able to capture fine visual features better than conventional CNNs, especially when classifying visually similar categories. It has achieved the best accuracy compared to other models in this experiment.

Model Fine-Tuning Strategy

All the models used in the Model Zoo have been trained using domain-specific fine-tuning methods. Specifically, this entails:

- Freezing of some initial layers for retaining general information
- Training of deeper layers for learning jewelry-specific information
- Specialized classification heads for learning eight output classes
- Hyperparameter tuning (learning rate, batch size, epochs)

In this way, we ensure that each model is optimally tuned towards the target domain.

Comparative Role of Model

The inclusion of multiple architectures within the Model Zoo enables a comprehensive comparative analysis. By exploring different design strategies, it becomes possible to identify the most suitable architecture for the jewelry classification task.
Through this evaluation, a balance between model accuracy and complexity is achieved, with results indicating that EfficientNetB2 performs the best for jewelry classification.

4) Training and Validation Engine

The role of the Training and Validation Engine is to oversee the entire learning procedure of all the deep learning models operating within the JewelNet architecture. The engine will be tasked with training the model effectively, validating its performance, and optimizing the model to ensure high generalization capabilities.
The dataset $D = (I_i, y_i)_{i=1}^{N}$ is split into three disjoint subsets to facilitate supervised learning:

- **Training Data (70%):** Data set used for training purposes.
- **Validation Data (15%):** Data set used for tuning the hyper parameters of the algorithm.
- **Testing Data (15%):** Data set used for testing the performance of the model.

The above strategy for partitioning the data set avoids the bias in the performance measure caused by data leakage.

Training Process

The training procedure consists of optimizing the model's parameters θ via mini-batch gradient descent. This consists of two stages: First, the model makes predictions for the current batch of training examples through forward propagation, and then updates its parameters based on the loss function through backpropagation.
The goal of training is to minimize the categorical cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{|\mathcal{C}|} y_{i,c}\log(\hat{y}_{i,c})$$

Here $y_{i,c}$ represents the true label, while $\hat{y}_{i,c}$ represents the predicted probability for class c.
The model parameters are updated through Adam optimization, which is a combination of adaptive learning rate and momentum:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta \mathcal{L}$$

where η is the learning rate and $\nabla_\theta \mathcal{L}$ represents the gradient of the loss function.

Optimization Strategies

Several optimization methods are adopted to guarantee efficient training processes as follows:

- **Adaptive Learning Rate Schedule:** Dynamic learning rate tuning helps accelerate learning in the early stages of the process, while decreasing learning rates helps in fine-tuning the model parameters.
- **Mini-Batch Learning:** Learning processes are conducted by splitting training sets into smaller mini-batches (batch size = 32).
- **Weight Initialization:** Transfer learning-based models adopt pre-trained weights, whereas the CNN model adopts custom weight initializations.

Regularization and Overfitting Control

Due to the small dataset size and complexity of deep learning models, overfitting becomes a critical problem. The following strategies are used to enhance generalization performance:

- **Early Stopping:** Training stops when validation loss ceases to improve, avoiding unnecessary overfitting.
- **Dropout:** Randomly disables neurons in the network during training.
- **L2 Regularization:** Adds a penalty term that favors smaller weight values.
- **Data Augmentation:** Applied at the preprocessing stage to introduce more variation in the data.

Validation and Model Selection

Since the training dataset is relatively small in size and deep learning models have a high complexity, overfitting is a common issue. The following measures have been implemented for mitigating overfitting:

- **Early Stopping:** The training process stops once the validation loss begins deteriorating.
- **Dropout:** This technique randomly turns off neurons during the training process.
- **L2 Regularization:** Large weights are penalized to simplify the model.
- **Data Augmentation:** This measure has been taken during the preprocessing phase.

Training Configuration

Training is carried out using the parameters outlined below:

- Number of epochs: up to 50
- Batch size: 32
- Optimizer: Adam
- Loss function: Categorical Cross-Entropy

These parameters are determined empirically in order to reach an optimal trade-off between training time and model performance.

Generalization and Stability

The process of splitting data, utilizing regularizers, and validating the model ensures the generalization capability of the trained model when applied to unseen data. This is particularly important in jewelry recognition because slight changes in visual characteristics can greatly influence the results obtained.
To conclude, the Training and Validation Engine provides a reliable system for training models.

5) Evaluation Module

The Evaluation Module provides a structured method for assessing the performance of all models within the JewelNet architecture. Rather than relying solely on overall accuracy, it uses a range of metrics to measure how well the model performs in terms of correct predictions, class-wise classification, and the distribution of errors. This type of evaluation is particularly important in fine-grained tasks such as jewelry classification.
Let the results of classification be represented by:

- True Positive (TP): Correct predictions that belong to the positive class
- True Negative (TN): Correct predictions that belong to the negative class
- False Positive (FP): Wrong predictions belonging to the positive class
- False Negative (FN): Wrong predictions belonging to the negative class

The following evaluations will be made based on these values.

Performance Metrics

The primary evaluation metric is **accuracy**, defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Although accuracy serves as an overall indicator of correctness, it does not necessarily indicate performance when working with multiple classes or imbalanced data sets. Thus, other indicators are used as well.
Precision indicates the share of correct positive predictions:

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

Recall (or sensitivity) measures the ability of the model to correctly identify positive instances:

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

Recall (also known as sensitivity) is the capacity of the algorithm to detect positive examples:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Additionally, error-specific metrics are computed:

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}$$

$$\text{False Negative Rate (FNR)} = \frac{FN}{FN + TP}$$

Such measurements give insight into the various forms of errors, which proves valuable when determining where a model fails.

Confusion Matrix Analysis

In order to conduct class-wise performance assessment, confusion matrices are created for each classifier. The confusion matrix is defined by $M \in R^{|C|\times|C|}$ and contains information about predicted vs actual class labels, with $M_{i,j}$ denoting the number of items with class i, but classified as j.
Confusion matrices help recognize classification problems for visually similar jewelry classes, such as:

- chains/necklaces, due to a similar elongated form;
- bangles/bracelets, due to a common circular shape;
- rings/ornaments, due to the similarity in size.

This allows us to see where the classifiers have problems and how they could be improved.

Per-Class Performance Evaluation

Alongside general metrics, there is also evaluation on the basis of per-class results. These include the calculation of precision, recall, and F1-score of each particular class. Such a metric as per-class evaluation is essential when it comes to proper classification, because not all classes can be classified equally well.
It is important because we may identify:

- Classes where there is a lot of misclassification
- Class bias
- Visual similarities

Interpretation of Results

Employing several evaluation criteria along with confusion matrix analysis enables a more comprehensive insight into the performance of a model. In comparison to the use of just one criterion, this approach ensures a general review of the model's functioning, highlighting both strengths and weaknesses.
From the JewelNet model design standpoint, the use of this technique ensures a fair comparison among different network architectures such as Custom CNN, VGG16, ResNet50, and EfficientNetB2. It can also provide proof of effectiveness of complex architectures in solving fine-grained classification problems.

Significance for Fine-Grained Classification

In case of classifying the jewelry with slightly similar visuals, there is need for high accuracy in classification. With this regard, there may be chances of overlooking the classification errors while evaluating through accuracy metrics only. The use of multi-metric approach facilitates the detection of errors that may require further investigations.
Overall, Evaluation module is considered efficient and effective way for evaluating model performance.

6) Recommendation Engine

In addition to classification, JewelNet uses a **content-based recommendation system** that enhances the usability of this tool significantly. In comparison to the traditional classification technique, which generates only categorical outputs, the recommendation engine finds similar items based on their feature vectors. Therefore, this mechanism transforms the algorithm into a decision support one that can be used in practice, for example, for the generation of recommendations in online stores.
Content-based recommendation works with the help of feature vectors extracted using neural networks. For example, a neural network receives an input image I and generates a set of features:

$$\mathbf{f} = \phi(I; \theta)$$

where ϕ represents the feature extractor and θ are the learnable model parameters. These features are normally extracted from the penultimate layer of the network model and include a lot of semantic and structural information about the image.
A collection of feature embeddings is created for all the images in the dataset:

$$\mathcal{F}_{db} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$$

For the query image, the similarity between the feature vectors is calculated using **cosine similarity** method:

$$\text{sim}(\mathbf{f}_q, \mathbf{f}_i) = \frac{\mathbf{f}_q \cdot \mathbf{f}_i}{\|\mathbf{f}_q\| \cdot \|\mathbf{f}_i\|}$$

where $\mathbf{f}_q$ denotes the query feature vector and $\mathbf{f}_i$ denotes a feature vector in the database. Using this measure, the algorithm retrieves the K nearest neighbors as follows:

$$\mathcal{R} = \text{Top-}K\ (\text{sim}(\mathbf{f}_q, \mathcal{F}_{db}))$$

This allows efficient searching of similar looking jewelry objects despite their dissimilarity of class membership and slight changes in design.
More specifically, from a domain-specific point of view, this feature is especially useful in jewelry because people are generally interested in finding visually similar items rather than categorizing them into certain classes. This recommendation system algorithm takes into account many small design aspects including texture, reflection, and construction.
In general, this additional module adds new functionality to JewelNet, which can be used in various applications.

3.2 System Workflow

The workflow of the system explains the entire process of how JewelNet operates, from obtaining input images to the final output. Getting familiar with the workflow allows one to understand the relationship among various components of the system.
As illustrated in Fig. 2, the process is divided into two primary stages: **training phase** and **inference phase**.

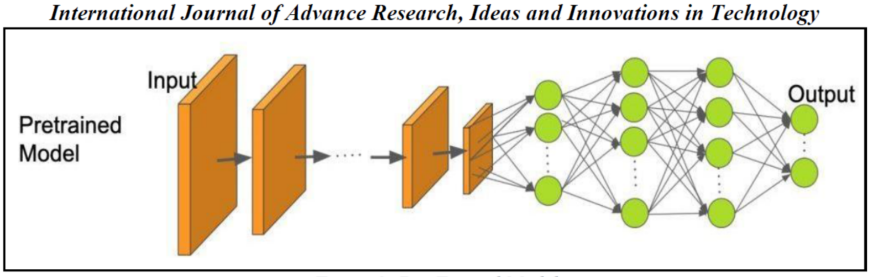


Figure 9: Pre-Trained Model

Figure 2: System workflow diagram from input to output.

Training Phase

Prior to the training phase, the raw jewelries undergo the process of quality filtering to eliminate blurry, corrupted, and mislabeled data. This is to ensure that the training data is accurate and consistent.
The processed images are subsequently fed into the preprocessing steps, where resizing is performed depending on the network requirements:

- 224×224 pixels for Custom CNN, VGG16, and ResNet50
- 260×260 pixels for EfficientNetB2

Normalization takes place using the standard values of 0 to 1 and ImageNet normalization. Augmentation is performed on the fly throughout the training process.
The data will be further divided into three sets: training (70%), validation (15%), and testing (15%). The training set will be utilized to update the model parameters via backpropagation, while the validation set is used for checking the performance of the model after an epoch.
Therefore, the training process could be described as:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}$$

where $\mathcal{L}$ is the loss function and η is the learning rate.
Early stopping is used when the validation performance does not improve, thus avoiding overfitting. Model weight values that perform well are stored through check pointing.

Inference Phase

For inference, an image is passed through the same preprocessing procedures (sans augmentation) before being passed through the pre-trained model. After that, class scores for each input are calculated using the Softmax function:

$$P(y = c \mid I) = \frac{e^{z_c}}{\sum_k e^{z_k}}$$

The predicted class is determined as:

$$\hat{y} = \arg\max_c P(y = c \mid I)$$

In addition to predicting the label, a confidence score can be given using the Softmax layer's output.

Recommendation Workflow

Concurrently, the features embedded from the input image are extracted and compared with those that have been pre-stored in the database using cosine similarity. The top K most similar products are then retrieved.
This dual-output pipeline enables the system to provide both:

- Accurate classification results
- Visually similar product recommendations

Overall Workflow Summary

The complete workflow of the JewelNet framework can be summarized as:

$$\text{Input Image} \rightarrow \text{Preprocessing} \rightarrow \text{Feature Extraction} \rightarrow \text{Classification} + \text{Recommendation}$$

Such an algorithmic structure ensures not only high precision of fine-grained classification but also the realization of different tasks required in practice.
In conclusion, the JewelNet system is a flexible and scalable solution for image analysis of jewelry items based on deep learning.

4 Research Methodology

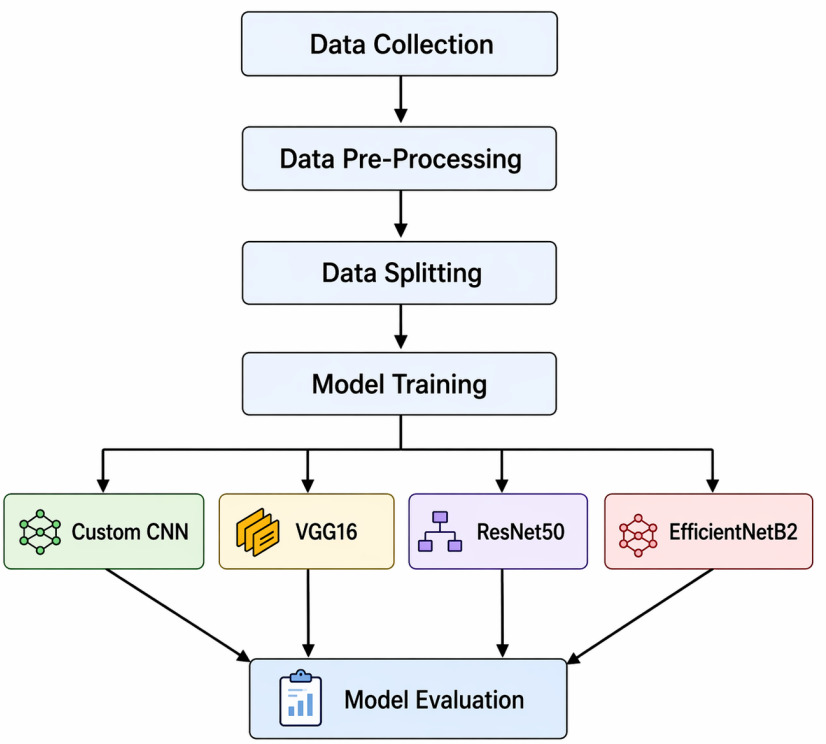


Figure 3: Data Processing and Multi-Model Training Workflow

As shown in Fig. 3, the proposed method follows a well-structured deep learning process for categorizing images belonging to jewelry with fine-grained attributes. The first phase of the deep learning workflow includes data gathering that takes place in realistic environments, followed by data processing, wherein images will be scaled, normalized, and augmented. Subsequently, this dataset will be split into three groups, namely training set, validation set, and test set. As the next step in model training phase, four different models (namely, Custom CNN, VGG16, ResNet50, and EfficientNetB2) are used to capture domain-specific and generalizable features.

4.1 Dataset Collection and Description

The procedure followed during the formation of the dataset ensured adequate diversity and quality to train efficient deep learning models for jewelry classification purposes. Since there is currently no publically available benchmarking dataset for this particular problem space, a dataset was created following a multi-source approach for its formation. The main sources utilized for collecting jewelry images include directly capturing images from authorized retail jewelry shops and local markets, obtaining images from publicly listed

products in e-commerce websites including those from various international e-commerce sites, and acquiring images from product catalogs with official permissions for their usage in this project.
Using this approach will ensure diversity in the type of images that will be used to train our model in terms of illumination, camera angle, and other such factors. This is necessary for allowing the model to learn feature representation which can generalize to unseen examples rather than fitting to an artificially homogenous distribution.
The dataset contains images belonging to eight different jewelry categories including:

$$\mathcal{C} = \{\text{Bangle}, \text{Bracelet}, \text{Chain}, \text{Earring}, \text{Necklace}, \text{Pendant}, \text{Ring}, \text{Nose Pin}\}$$

These categories were chosen not only to cover various types of jewelry but also to incorporate the challenge of having classes that would be structurally similar yet distinctive enough. For example, bangles and bracelets, as well as chains and necklaces, belong to categories whose representatives are very similar in structure and require the learning of subtle discriminatory features associated with shape, stiffness, and other patterns.
Each picture in the dataset was carefully verified and curated for the quality of its data and labels. The following rules have been implemented:

- Only pictures with a single identifiable piece of jewelry are allowed;
- Pictures with strong blurriness, noise, and/or unusual lighting are prohibited;
- Pictures with ambiguities regarding the label of the object depicted on them are prohibited;
- Duplicates and near-duplicates are prohibited.

In addition, the attempts have been undertaken to maintain the intra-class diversity of examples, thus allowing the model to learn general and discriminative features inherent in the objects. These features include different designs, sizes, materials used (e.g., gold, silver, platinum, gemstone-encrusted), etc.
This dataset may be represented formally as:

$$D = \{(I_i, y_i)\}_{i=1}^{N}$$

Here, I_i refers to the i-th image, and $y_i \in \mathcal{C}$ is its class label. The dataset consists of $N = 1,217$ images.
In order to perform supervised learning as well as unbiased evaluation, a train-validation-test split was performed on the data using a stratified sampling technique:

- Training set: 70% (851 images)
- Validation set: 15% (181 images)
- Test set: 15% (185 images)

This partition guarantees that each partition has an adequate representation of all classes, thus ensuring proper model training and testing.
As seen in Table 4.1, the dataset contains almost equally distributed classes among each category, with only minor changes caused by practical limitations associated with data acquisition. Equal distribution is important for avoiding the potential bias of the model towards any particular category.

Table 4.1: Dataset Distribution Before and After Augmentation

Category	Original	Train (70%)	Val (15%)	Test (15%)	Augmented
Bangle	140	98	21	21	980
Bracelet	147	103	22	22	1,029
Chain	153	107	23	23	1,071
Earring	163	114	24	25	1,141
Necklace	162	113	24	25	1,134
Pendant	150	105	22	23	1,050
Ring	152	106	23	23	1,064
Nose Pin	150	105	22	23	1,050
Total	**1,217**	**851**	**181**	**185**	**8,519**

To enrich the dataset with more images, data augmentation was implemented on the training dataset. Consequently, the augmented training dataset consists of 8,519 images instead of only 851. The process of data augmentation enhances the generalization ability of the model in predicting the classes because new data simulates variations of scale, orientation, and lighting.
The test subset, which consists of 185 images, constitutes around 15 percent of the entire dataset and does not overlap with the training and validation datasets at all.
Thus, the constructed dataset can serve as a reliable basis for developing fine-grained jewelry image classification models.

4.2 Data Preprocessing and Augmentation

There exists high variance among images of jewelry due to variations in spatial resolution, lightening conditions, background complexity, and pose. This might adversely affect the stability and convergence properties of deep learning models. Therefore, we apply a correct preprocessing technique to ensure that the images are processed in such a way that effective feature learning can be achieved.
The proposed preprocessing technique comprises three sequential steps. The first step includes resizing the image, normalization of pixel values, and then data augmentation.

4.2.1 Image Resizing

Let us denote an input image as $I \in \%mathbbR^{H \times W \times C}$, where H, W, and C represent height, width, and channel numbers, respectively. Given that images in the data set have been sourced from different locations, their dimensions are quite disparate. Hence, resizing all images to standardized dimensions becomes necessary so as to allow compatibility with the CNN model.
The VGG16, ResNet50, and Custom CNN models demand input images of the following dimensions:

$$I' \in R^{224 \times 224 \times 3}$$

while EfficientNetB2 uses a higher resolution:

$$I' \in R^{260 \times 260 \times 3}$$

in line with its compound scaling structure.
The technique uses bilinear interpolation for upsizing while employing area interpolation for downsizing. By applying such methods, it is ensured that the process of resizing helps in retaining spatial details without increasing the computational overhead.
Besides, resizing leads to having a consistent receptive field for all models to have a fair comparison among them.

4.2.2 Normalization

After resizing, the intensities of pixels are normalized for computational convenience and faster convergence in the learning phase. The pixel intensities for the Custom CNN, ranging between $[0, 255]$, are normalized to:

$$I'' = \frac{I'}{255}$$

Thus, the normalized range is obtained in the interval of $[0, 1]$.
In transfer learning models like VGG16 and ResNet50, images are standardized based on the statistics of ImageNet. This process includes the following steps:

$$I''' = \frac{I' - \mu}{\sigma}$$

Where μ and σ stand for the mean and standard deviation that have been calculated based on the ImageNet dataset.
This method is used to bring the input distribution into alignment with the pre-training distribution, thus enabling effective reuse of the learnt weights during the fine tuning process. It also ensures stability in the updates to the gradients to prevent their explosion/vanishing.

4.2.3 Data Augmentation

Due to the rather small number of examples present within the training set, data augmentation is utilized as an important step in order to achieve higher generalization power and avoid overfitting on the training data.
Mathematically, a transformation procedure is carried out using an augmentation function $\mathcal{A}(\cdot)$ as follows:

$$I_{aug} = \mathcal{A}(I'')$$

where $\mathcal{A}$ denotes a combination of geometric and photometric transformations.
Augmentation procedures have been developed to mimic variations that occur in real-life scenarios with respect to jewelry images. The type of augmentation applied is described in Table 4.2.
The application of these augmentations is random, meaning that the training will see various instances of the dataset at each epoch, making sure that the model does not memorize any particular features of the images but learns to extract invariant and discriminative features instead.
When thinking about the use of augmentations on the level of domain, augmentation is a necessity when classifying jewelry items due to the presence of metal components and stones. A very small change in the lighting or position of the camera can drastically affect how a jewelry item looks, and adding such variation during training will increase model robustness.
Moreover, using augmentation enables us to increase the training sample size without collecting new data.

Table 4.2: Data Augmentation Techniques Applied

Technique	**Parameter Range**	**Purpose**
Rotation	±20° to ±25°	Viewpoint invariance
Width Shift	±20%	Horizontal position invariance
Height Shift	±20%	Vertical position invariance
Shear Transform	±0.15 to ±0.2	Shape distortion tolerance
Zoom	±20% to ±30%	Scale invariance
Horizontal Flip	Enabled	Mirror symmetry learning
Brightness Adjust	0.8 to 1.2 range	Lighting condition variance
Channel Shift	±10%	Color variation tolerance
Fill Mode	Nearest neighbor	Boundary pixel handling

4.3 Custom CNN Architecture

The Custom CNN is designed to suit the jewelry recognition task because it incorporates architectural design that is influenced by the characteristics of the jewelry images, such as large intraclass variation, overlapping classes, and reflections. The Custom CNN differs from generic CNN designs in that it aims to detect discriminating features such as textures, edges, and reflections that help differentiate visually similar jewelry classes.

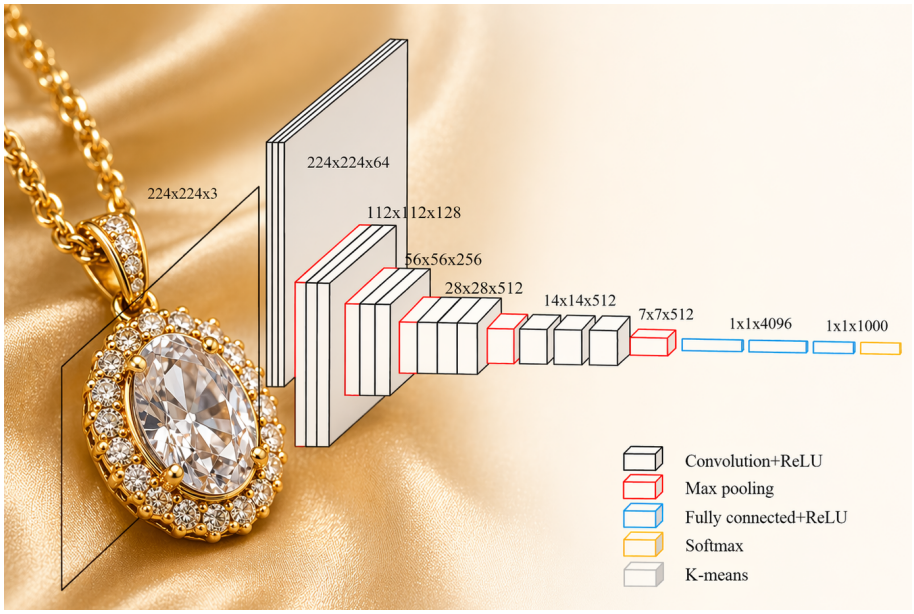


Figure 4: Custom CNN architecture with layer-wise configuration

As illustrated in Fig. 4, The architecture of the neural network follows a **progressive hierarchy-based feature learning** mechanism. This means that each layer learns more advanced features based on the features present in its inputs. Simple features such as edges and texture features are learned by the early layers, while advanced features related to the object's design are learned by the deep layers.
The basic computation performed in each layer is as follows:

$$(I * K)(x, y) = \sum_{m} \sum_{n} I(x - m, y - n)\, K(m, n) \tag{4.1}$$

where I denotes the input feature map, K is the convolution kernel which could be learned,

while (x, y) represents the location in the output feature map. This would allow the neural network to capture local features using weight sharing and receptive field extension.
All the convolution layers make use of 3×3 kernels with proper padding, along with a Rectified Linear Unit (ReLU) activation function:

$$f(x) = \max(0, x)$$

This method ensures non-linear behavior and prevents the vanishing gradient issue.
The network is made up of four convolutional layers that are succeeded by batch normalization and max pooling. The number of filters used gradually increases from 32 to 64, 128, and 256. This makes it possible for the model to learn high-level abstraction. The max pooling operation reduces dimensionality while keeping only the essential elements Table 4.3.

Table 4.3: Custom CNN Layer-wise Configuration

Block	**Layer Configuration**	**Filters/Units**	**Output Shape**	**Purpose**
Input	Raw image input	—	$224 \times 224 \times 3$	Input layer
Block 1	Conv2D + BatchNorm + MaxPool 2×2	32	$111 \times 111 \times 32$	Edge & texture detection
Block 2	Conv2D + BatchNorm + MaxPool 2×2	64	$54 \times 54 \times 64$	Local pattern learning
Block 3	Conv2D + BatchNorm + MaxPool 2×2	128	$26 \times 26 \times 128$	Part-level feature extraction
Block 4	Conv2D + BatchNorm + MaxPool 2×2	256	$12 \times 12 \times 256$	High-level semantic encoding
FC1	Flatten + Dense + Dropout(0.5)	512	512	Global feature representation
FC2	Dense + Dropout(0.3)	256	256	Feature refinement
Output	**Dense + Softmax**	**8**	**8**	**Class probability output**

Batch normalization is used after each conv layer to normalize the activations' mean to be close to 0 and variance to be around 1. It ensures stability in the training process by decreasing internal covariate shift.
Dropout regularization is used on the fc layers with 0.5 and 0.3 probability, respectively. The technique ensures that neurons are deactivated randomly to avoid overfitting. This method is crucial since the dataset size is relatively small.
In addition, **L2 weight regularization** is used on all layers with a 0.01 regularization coefficient to ensure that weights remain within an appropriate range.

$$\mathcal{L}_{reg} = \lambda \sum_{i} w_i^2$$

where λ is the regularization parameter. It promotes simplicity in models and enhances generalization.
The dense layers act as aggregators of higher-level features that convert the spatial feature maps to a feature vector. The output layer, called the Softmax layer, returns the probabilities for the eight jewelry classes as follows:

$$P(y = c) = \frac{e^{z_c}}{\sum_{k=1}^{8} e^{z_k}}$$

where z_c is the logit of class c.
On the whole, the Custom CNN model structure is intended to be a model that balances model complexity and computation efficiency while being able to make the best use of fine details of jewelry images in its learning process.

4.4 Transfer Learning Models

Transfer learning has been used to benefit from the gained knowledge using large-scale datasets like ImageNet. Hence, it was possible to extract features effectively despite the relatively limited size of the dataset in the studied domain. With the help of transfer learning, pre-trained models were able to benefit from the previously gained experience regarding low-level and high-level visual features that resulted in a faster rate of convergence and improved performance. In this work, three deep convolutions were used: **VGG16**, **ResNet50**, and **EfficientNetB2**. Such different approaches for architecture have been chosen to ensure a diverse range of comparisons and results considering differences in depth, complexity, and computation power required. **VGG16** is the deep CNN with 16 weight layers, which are organized as 3×3 convolution filters in a sequential way. Thus, it is capable of extracting hierarchical features at low and high levels of abstraction. Due to its simple architecture and efficient feature representation, VGG16 could be considered one of the most reliable approaches to transfer learning. **ResNet50** uses residual learning to facilitate gradient propagation through layers with the help of skip connections. The residual mapping is given by the formula:

$$y = F(x) + x$$

where x is the input, while $F(x)$ is the learned residual function. The model addresses the vanishing gradient issue and facilitates the construction of deeper neural networks. Another modern network, **EfficientNetB2**, applies a strategy called compound scaling, where the depth, width, and resolution of the network inputs are scaled concurrently:

$$\text{depth} = \alpha^{\phi}, \quad \text{width} = \beta^{\phi}, \quad \text{resolution} = \gamma^{\phi}$$

The efficiency of this well-balanced scaling strategy helps EfficientNetB2 to be more accurate while using less number of parameters and less computational cost in comparison with traditional CNN designs. It is also important that EfficientNetB2 uses MBConv and SE blocks in order to improve feature extraction and channel-wise attention respectively. In the suggested model, all pretrained models were fine-tuned by freezing the convolutional layers and further training higher layers with the use of the jewelry dataset. Special classification layers were included into all of these networks in order to classify the jewelry dataset to 8 specific classes.

4.4.1 VGG16

The VGG16 [56] neural network is an architecture which consists of 16 layers of learned weights, consisting of five blocks of convolutional layers with three fully connected layers at the end. This architecture is defined by its use of small (3×3) convolutional kernels with a stride of 1 that allow for efficient deep learning of hierarchical features.
In this model, the convolutional blocks become deeper over time, which helps the model to learn more complex representations. Low-level features (such as edge information and texture information) can be learned in early layers of the network, whereas higher-level features like shape information and pattern information will be learned by deeper layers of the network. This helps to capture fine-grained features, which is very useful in jewelry recognition tasks.
For adapting this model to our specific task, the original classification head, designed for 1000 classes from ImageNet, was removed and replaced by a new classification layer for our jewelry dataset. Specifically, the new classification layer used by the model consisted of the following:

- Global Average Pooling (GAP)
- Dense layer with 512 units and ReLU activation
- Dropout layer with rate 0.3
- Dense layer with 256 units and ReLU activation
- Dropout layer with rate 0.3
- Output layer with Softmax activation for 8 classes

The Global Average Pooling layer shrinks the spatial size of the feature maps without losing information in terms of channels, resulting in fewer parameters than fully connected layers and avoiding overfitting.
The Softmax output layer calculates the probability of each class as follows:

$$P(y = c) = \frac{e^{z_c}}{\sum_{k=1}^{8} e^{z_k}}$$

where z_c denotes the logit associated with the c^{th} class.

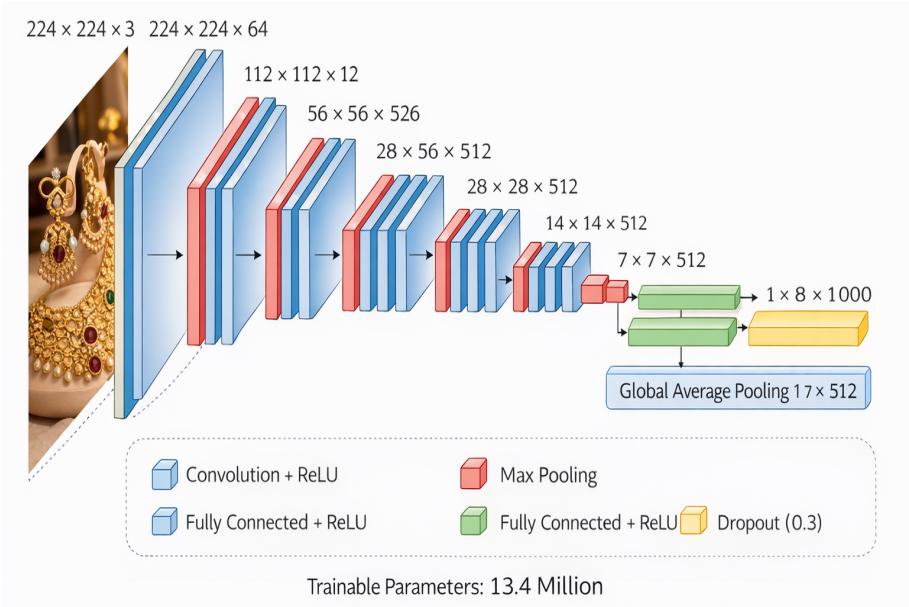


Figure 5: Architecture of the VGG16-based transfer learning model used for jewelry classification.

As depicted in Fig. 5, the modified architecture utilizes the robust feature extraction properties of the VGG16 network alongside a lighter and custom classification head that is optimized for the jewelry dataset.
The fine-tuning process involved the implementation of a two-phase training approach to successfully modify the pre-trained model while retaining its learned features:

- **Stage 1 (Feature Extraction Phase):** n this phase, all the pre-trained layers of the VGG16 network were frozen and only the classification head was fine-tuned. This ensures that the model is able to learn the decision boundaries specific to the current task without modifying any previously learned features on ImageNet.

- **Stage 2 (Fine-Tuning Phase):** In the second phase, the last 8 layers of the VGG16 base network were unfrozen and fine-tuned along with the classification head. To prevent catastrophic forgetting, which is the phenomenon of overwriting of previously learned features, a smaller learning rate was employed during this phase.

The learning rate scheduler helps ensure convergence stability in fine-tuning. The lower layer parameters will be kept constant, allowing the model to retain its general vision ability, whereas the upper layer parameters will be adapted to specific domain attributes such as texture, reflectivity, and structure.
The dropout technique will help minimize overfitting in fully connected layers, whereas global average pooling will keep the number of parameters low and enhance generalization. In conclusion, the VGG16 architecture with transfer learning offers an excellent starting point for fine-grained jewelry recognition because of its superior hierarchical feature extraction capabilities and effective adaptation using fine-tuning.

4.4.2 ResNet50

ResNet50 [24] is a type of deep CNN that implements residual learning to allow for training very deep architectures. The issue with deep architectures is the vanishing gradient problem since the gradient becomes smaller as it backpropagates through time. The ResNet50 architecture resolves this issue by implementing identity mappings. The residual learning function is formulated as follows:

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{4.2}$$

where $\mathcal{F}(x, W_i)$ is the residual mapping learned by the stacked convolutional layers and x is the identity shortcut connection. Such formulation allows the model to learn residual functions rather than direct mappings, enabling gradient propagation and training of deeper networks.
The ResNet50 network is structured to include 50 layers divided into several residual blocks, with each residual block having a sequence of convolutional layers followed by batch normalization and rectified linear unit (ReLU) activation function. The hierarchical structure of the network makes learning of feature representation possible, starting from low-level details such as edges and texture to higher-level information related to object structure and categories.
In the context of jewelry image classification, the last fully-connected layer in ResNet50, initially designed for 1000 categories of ImageNet, is discarded. Instead, the output feature maps after the final residual block are processed using **Global Average Pooling (GAP)** resulting in a smaller feature map size that produces a feature vector:

$$\mathbf{f} \in R^{2048}$$

The following feature vector of 2048 dimensions contains semantic data extracted from the input image.
A custom classifier head is utilized, much like the architecture of the VGG16-based classifier, where the layers include:

- Dense layer of size 512 with ReLU activation
- Dropout layer with a dropout value of 0.3
- Dense layer of size 256 with ReLU activation

- Dropout layer with a dropout value of 0.3
- Output layer with Softmax activation for 8 target classes

Softmax layer outputs the probability values over the target classes.

$$P(y = c) = \frac{e^{z_c}}{\sum_{k=1}^{8} e^{z_k}}$$

where z_c is the logit corresponding to class c.

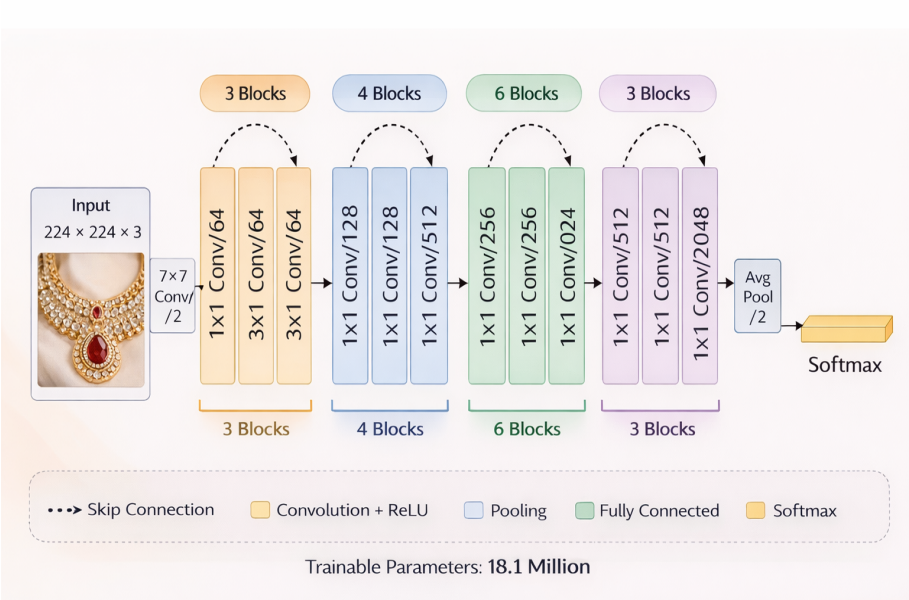


Figure 6: Architecture of the ResNet50-based transfer learning model used for jewelry classification.

As illustrated in Fig. 6, the architecture integrates deep residual learning for feature extraction, complemented by an efficient classification head designed specifically for the jewelry dataset.
During fine-tuning, a selective freezing approach was employed. Most of the lower layers were kept frozen to preserve the general features learned from ImageNet, while some of the higher layers remained trainable to adapt to domain-specific characteristics. Specifically, all layers except the final 50 were frozen throughout training.
This approach maintains general visual representations while enabling deeper layers to learn features relevant to jewelry classification, such as variations in shape, structure, and reflective properties.
A small learning rate was used during fine-tuning to ensure stable convergence without significantly modifying the pre-trained weights. In addition, dropout regularization was applied to the classification head to reduce the risk of overfitting, particularly given the limited dataset size.
Although ResNet50 is a deep model with strong feature extraction capabilities, it can be sensitive to the size and complexity of the dataset. However, its use of residual learning makes it a suitable baseline for comparison within the JewelNet framework.

4.5 EfficientNetB2 Model

EfficientNetB2 is an example of a modern CNN architecture that achieves greater accuracy with efficient computation through the method of **compound scaling**. This model differs

from older versions that individually scale each of the three components by using compound scaling for all three components relative to the base model (EfficientNetB0).
Compound scaling equation is stated as follows:

$$\text{depth} = \alpha^{\phi}, \quad \text{width} = \beta^{\phi}, \quad \text{resolution} = \gamma^{\phi}$$

where $\alpha = 1.2$, $\beta = 1.1$, and $\gamma = 1.15$ are scaling factors, and ϕ is the compounded scaling factor. This careful balance in scaling allows EfficientNetB2 to deliver outstanding results while remaining computationally efficient.
The network consists of several **Mobile Inverted Bottleneck Convolution (MBConv)** layers. MBConv layers employ depthwise separable convolutions in order to decrease the amount of parameters used and computational costs. An MBConv layer consists of expansion of input channels, application of depthwise convolution, and dimensionality reduction using a pointwise convolution.
One of the important elements of EfficientNetB2 is the use of the **Squeeze-and-Excitation (SE)** mechanism in each MBConv layer. It allows modeling inter-channel dependencies using the channel-wise attention mechanism. It works as follows:

- **Squeeze:** The global average pooling operation is performed on the feature maps to generate the channel descriptor:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_c(i, j)$$

- **Excitation:** The excitation descriptor is processed using two fully connected layers with a ratio of 4:

$$s = \sigma(W_2 \cdot \delta(W_1 \cdot z))$$

  Where δ is the Rectified Linear Unit activation function, and σ is the Sigmoid activation function.

- **Recalibration:** The generated attention weights are used to multiply the

$$\tilde{X}_c = s_c \cdot X_c$$

The benefit of such a procedure is that the system will concentrate on significant features while discarding irrelevant ones, which makes this approach particularly useful in classification problems, especially those related to identifying jewelry.
As shown in Fig. 7, the EfficientNetB2 network hierarchically captures features using stacks of MBConv blocks, after which it aggregates features globally.
In our case of jewelry image classification, we substituted the classification head of the standard EfficientNetB2 network with a customized classification head that is best suited to the data. The customized classification head structure includes:

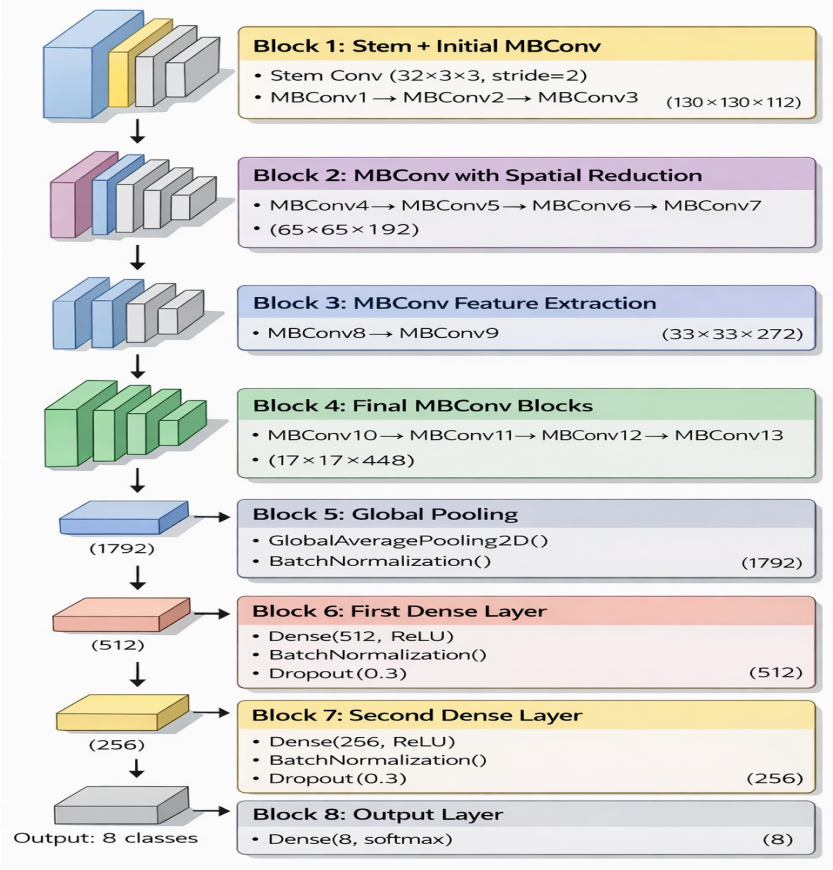


Figure 7: EfficientNetB2 architecture with compound scaling

- GAP layer to flatten spatial dimensions
- Batch Normalization to normalize activations
- Dense layer with 512 neurons, ReLU activation, and L2 regularization ($\lambda = 0.01$)
- Dropout layer with dropout rate 0.5
- Dense layer with 256 neurons and ReLU activation
- Dropout layer with dropout rate 0.3
- Softmax output layer with 8 classes

The Softmax layer calculates class scores using the following formula:

$$P(y = c) = \frac{e^{z_c}}{\sum_{k=1}^{8} e^{z_k}}$$

where z_c is the logit corresponding to class c.
Fine-tuning is performed using the following partial training technique: in order to benefit from the information about the common features of different objects contained in the pre-trained EfficientNetB2 network, the last 50 layers of the architecture are left trainable while all preceding layers are frozen.
The lower learning rate allows for better stability and avoids forgetting pre-trained weights during the tuning process. Dropout and L2 regularization methods have been used for better generalization and to avoid overfitting.

EfficientNetB2 performs significantly better in the task of fine jewelry classification because of its ability to extract more detailed features and pay more attention to important characteristics. Its efficient scaling strategy and channel-wise attention make this architecture especially effective at detecting visually close items like chains and necklaces or bangles and bracelets.
On the whole, the EfficientNetB2 architecture is the most sophisticated in the JewelNet framework because it provides the best classification performance without sacrificing computational efficiency.

4.6 Training Protocol

Training was performed in a controlled manner for all four architectures based on a common training protocol, which allows for an unbiased comparison by controlling for any bias due to differences in the training methods used.
The objective function is based on the minimization of the categorical cross-entropy loss, which can be mathematically formulated as:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c})$$

where $y_{i,c}$ indicates the true label, while $\hat{y}_{i,c}$ stands for the predicted probability of class c. This cost function is ideal for multi-class classification problems since it gives a probabilistic view of the predictions made by the model.
Adam optimizer was used because of its dynamic learning rate approach, which integrates the benefits of both momentum and RMSProp optimization algorithms. The update formula is:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

where $\hat{m}_t$ and $\hat{v}_t$ are corrected unbiased first and second moment estimates respectively, and η stands for the learning rate. An initial learning rate value of 10^{-4} was chosen, along with a learning rate decay factor per update of 10^{-6}.
A **learning rate decay strategy** was also utilized to further stabilize the process. Whenever there was no increase in the validation loss value for 5 consecutive epochs, the current learning rate was multiplied by 0.5, allowing fine-tuned training updates in later stages, as well as preventing oscillations near local minima.
Training is done using mini-batches of 32 examples to allow for both efficiency and effective convergence. The maximum number of epochs is set to be 50; however, training usually ends sooner due to early stopping.
Early stopping criterion involves monitoring validation accuracy and stopping training if there is no improvement for 10 consecutive epochs in order to avoid overfitting.
Additionally, **model checkpointing**, based on validation accuracy, is performed in order to keep the most accurate model obtained in case the performance begins to decline later on during training.
Taking into account the small class imbalance observed in the training data, **class weights** are calculated and applied during training, ensuring proportional contribution of all the classes to the loss function.
The choice of input image dimensions depends on requirements of the model architecture. Input dimensions for VGG16, ResNet50 and Custom CNN models are:

$$224 \times 224 \times 3$$

while EfficientNetB2 uses:

$$260 \times 260 \times 3$$

to align with its compound scaling design.

Table 4.4: Training Configuration for All Models

Hyperparameter	**Value**
Optimizer	Adam (lr=10^{-4}, decay=10^{-6})
Loss Function	Categorical Cross-Entropy
Batch Size	32
Maximum Epochs	50
Early Stopping	Patience = 10 (monitor: val_accuracy)
LR Reduction	Patience = 5, factor = 0.5
Model Checkpoint	Save best validation accuracy
Input Size (VGG/ResNet/CNN)	$224 \times 224 \times 3$
Input Size (EfficientNetB2)	$260 \times 260 \times 3$
Class Weights	Computed to handle minor class imbalance

As Table 4.4 ,the use of adaptive optimization algorithms, learning rate schedules, early stopping, and regularization helps achieve stable convergence and high generalization capabilities for all networks. Such a well-thought-out model training procedure allows for a fair comparison between the networks under consideration.

4.7 Evaluation Metrics

The performance assessment of classification models demands multiple criteria, which will give us an opportunity to evaluate their performance in different ways. In the case of fine classification, like jewelry classification, using one criterion (for example, accuracy) can lead us to misunderstanding the performance of certain classes and misclassification of data. Thus, a multi-criteria approach is necessary, which will help us make not only quantitative but also qualitative conclusions about the performance of the model.
Denote the values for TP (true positive), TN (true negative), FP (false positive), and FN (false negative) as the parameters used in the calculation of performance metrics.
The next metrics are calculated for each model under consideration in two ways (on an overall basis and per class):

- **Accuracy:** Is the metric of correctly classified samples' total percentage in relation to all instances:
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
It provides an overall assessment of classification's quality but does not consider individual class characteristics and may be misleading for imbalanced data sets.

- **Precision:** Is the indicator showing what proportion of predicted positives is classified correctly in relation to all predictions:
$$\text{Precision} = \frac{TP}{TP + FP}$$

High precision value suggests low probability of making false positive predictions and is crucial in tasks requiring high precision.

- **Recall (Sensitivity):** Measures what part of real positive samples is predicted properly:
$$\text{Recall} = \frac{TP}{TP + FN}$$
High value means that the vast majority of real instances are identified, thus minimizing false negative errors.

- **F1-Score:** Represents the harmonic average of precision and recall:
$$F_1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
It helps to balance between precision and recall in case of their conflict.

- **False Positive Rate (FPR):** Indicates how many negative examples are mistakenly classified as positive in percent:
$$\text{FPR} = \frac{FP}{FP + TN}$$
Low FPR shows higher reliability of the classifier in terms of avoiding false alarms.

- **False Negative Rate (FNR):** Shows how many positive instances are predicted incorrectly:
$$\text{FNR} = \frac{FN}{FN + TP}$$
Low FNR value minimizes false negatives and is vital in the analysis of positive cases.

- **Confusion Matrix:** Square matrix of size $M \in R^{C \times C}$, where $M_{i,j}$ denotes the number of instances from class i predicted as class j.

In the case of multiclass classification, these metrics are calculated using a one-vs-all scheme for every class and subsequently averaged across classes by means of macro or weighted averaging.
A special place in fine jewelry classification is taken by the confusion matrix since it allows identifying the patterns of model mistakes. For instance, it might be that chain and necklace or bangle and bracelet tend to be confused more often because of their resemblance. This would help to see the weaknesses of a model and improve it in the future.
Thus, employing different evaluation metrics allows conducting a holistic assessment of the model, taking into account both its general effectiveness and peculiarities of class recognition.

5 Experimental Results and Analysis

5.1 Experimental Setup

All experiments were performed under the same computational conditions to guarantee the reliability of our results. Careful consideration was taken to design an unbiased experiment by avoiding the biases that can be induced by data splitting, model initialization, and training.
The dataset $D = (I_i, y_i)_{i=1}^{N}$, which consists of $N = 1,217$ labeled jewelry pictures belonging to eight categories, was divided into three disjoint sets through stratified sampling:

- Training set: 70% (851 images)
- Validation set: 15% (181 images)
- Test set: 15% (185 images)

Stratified sampling ensures that each strata is representative of all classes, eliminating any possibility of class distribution bias, which allows for proper assessment of the model.
In order to achieve consistency and reproducibility of the experiment, random seeds were set at every step in the process pipeline, from splitting of data to initialization of weights and batch shuffling.
Models were trained according to one training scheme with the same set of parameters, as described in Chapter 4 Table 4.4. In this way, the performance variance between models will result from their architectural design rather than training process variation.
The process of training consists of learning a mapping function:

$$f_\theta : I \rightarrow \hat{y}$$

where I stands for the input image, $\hat{y}$ is the output class label prediction, and θ refers to the optimized model parameters throughout the training process.
The test set remained strictly independent from the training and validation phases, being used only once for evaluating the model's performance. Such a division avoids data leaks and guarantees a fair estimation of the model's generalization ability.
The performance evaluation was done based on 185 test images, belonging to all eight jewel categories. The Softmax function is applied to each test image by the trained model as follows:

$$P(y = c \mid I) = \frac{e^{z_c}}{\sum_{k=1}^{8} e^{z_k}}$$

where z_c denotes the logit for class c. The output class label can be found by:

$$\hat{y} = \arg\max_c P(y = c \mid I)$$

Evaluation measures were calculated based on both per-class and combined approaches. In case of multiclass evaluation, the one-vs-all approach was applied to calculate True Positives, False Positives, True Negatives, and False Negatives of each class. Then, these measures were used to derive performance measures including precision, recall, and F1-score.
In order to ensure an exhaustive evaluation of each model, two types of metrics were considered: **macro-averaged** and **per-class**. Macro-averaged metrics represent the arithmetic average of metrics taken across all classes with equal weight given to each class. The presented experiment allows for conducting a proper evaluation of all models, which can be compared against each other regarding their ability to perform fine-grained classification of jewels.

5.2 Overall Performance Comparison

The comparative performance of the four models evaluated is given in Table 5.1. It can be clearly seen from the results that the deep learning model-based classification outperforms the traditional manual feature extraction methods used in previous studies by a significant margin. Thus, it is established that the convolutional models can effectively capture fine-grained features.
It is evident from Table 5.1 that all four models perform excellently well in classifying the jewelries, with an accuracy ranging between 87.97% and 95.21%. Although there is not much difference in terms of accuracy, however, when the other performance metrics are considered, it becomes apparent that the difference is noticeable.

Table 5.1: Overall Performance Comparison of All Models

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**FPR**	**FNR**
Custom CNN	93.78%	0.9398	0.9378	0.9381	0.0089	0.0622
ResNet50	87.97%	0.8826	0.8797	0.8799	0.0172	0.1202
VGG16	94.19%	0.9470	0.9419	0.9420	0.0083	0.0581
EfficientNetB2	**95.21%**	**0.9533**	**0.9506**	**0.9497**	**0.0075**	**0.0494**

Overall, out of all the models used, **EfficientNetB2** shows the highest performance on all criteria of evaluation. With 95.21% accuracy, 95.33% macro precision, 95.06% recall, and F1-score of 94.97%, it demonstrates a highly efficient performance in terms of discriminating between visually similar classes of jewelry. Low FPR (0.75%) and FNR (4.94%) of the model show that it has a well-balanced classification mechanism and minimizes misclassification errors.
The best performance of the EfficientNetB2 model is explained by its compound scaling approach and integration of the SE block modules. Simultaneous scaling of depth, width, and image resolution allows the network to extract richer feature representations. Moreover, channel-wise feature discrimination by means of the Squeeze-and-Excitation blocks helps improve the overall performance and classification efficiency.
The second-best performance is achieved by the VGG16 architecture, which reaches an accuracy score of 94.19%. High performance is achieved due to the model's deep and hierarchical structure allowing for successful feature extraction on various levels of abstraction. Low scores on FPR and FNR criteria also indicate a stable classification process with minimum error rates.
The custom convolutional neural network achieves comparable performance with 93.78% accuracy. The performance of the architecture is quite high considering that the model

was completely trained from scratch without any pre-trained layers. Good performance indicates the effectiveness of domain-specific training and ability to learn such class characteristics as texture patterns, shape, or reflectance. Higher FNR (6.22%) indicates that some true positives were not detected.
Lastly, **ResNet50** shows the lowest score for the accuracy of classification, reaching only 87.97%. Although the architecture makes use of residual connections, it fails to achieve high performance within the problem context. The poor performance of the architecture could be caused by relatively small data size, preventing proper adaptation of a deeper architecture.
Therefore, from a comparative analysis standpoint:

$$\text{EfficientNetB2} > \text{VGG16} > \text{Custom CNN} > \text{ResNet50}$$

As shown above, the results prove that state-of-the-art architectures with proper scalability and attention techniques work better than conventional deep learning models and specifically designed architectures.
All in all, the outcomes of our experiments indicate that EfficientNetB2 is the best choice in terms of accuracy and efficiency while achieving a balance between these factors.

5.3 Per-Class Performance: Custom CNN

The table below shows per-class performance of the Custom CNN model for all eight types of jewelry. It should be noted that per-class analysis gives more detailed information compared to overall accuracy, which makes it possible to find strengths and weaknesses of the model in recognizing fine-grained visual patterns.

Table 5.2: Per-Class Performance Metrics for Custom CNN

Class	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**FPR**	**FNR**
Bangle	98.34%	0.9062	0.9667	0.9355	0.0142	0.0333
Bracelet	99.59%	1.0000	0.9667	0.9831	0.0000	0.0333
Chain	97.51%	0.8750	0.9333	0.9032	0.0190	0.0667
Earring	98.76%	0.9655	0.9333	0.9492	0.0047	0.0667
Necklace	97.93%	0.9032	0.9333	0.9180	0.0142	0.0667
Nose Pin	98.34%	0.9355	0.9355	0.9355	0.0095	0.0645
Pendant	98.76%	1.0000	0.9000	0.9474	0.0000	0.1000
Ring	98.34%	0.9333	0.9333	0.9333	0.0095	0.0667
Macro Avg	**98.45%**	**0.9398**	**0.9378**	**0.9381**	**0.0089**	**0.0622**

As Table 5.2, The Custom CNN has shown consistent good performance in most of the classes, with a macro-average F1-score of 0.9381. Thus, this architecture proves its capacity for feature learning even when trained from scratch (no pre-trained weights used).
The model exhibits **perfect precision** for *bracelet* (1.000) and *pendant* (1.000), implying that all predictions for these classes are positive, i.e., there are no false positives at all. As it seems that this jewelry has quite distinct characteristics – bracelets are quite rigid in terms of construction, and pendant designs have unique shapes – the model could learn their distinguishable features.
The earrings, rings, and nose pins demonstrate decent F1-scores (exceeding 0.93), meaning that the Custom CNN successfully learns their distinguishable features (such as shape contours, symmetry, and sizes).

The model demonstrates worse performance for such classes as *chain* and *necklace*. These classes have almost similar characteristics – they are long objects of jewelry, with quite a lot of overlap. As it can be seen from the metrics (quite high FPR/FNR values), it leads to increased inter-class confusion.
The *pendant* class demonstrates great precision (1.000) but a quite high FNR (0.1000). It means that not all pendant instances were recognized correctly, although the class precision score indicates otherwise. In some cases, it could happen because of the pendant design similarity to other jewelry pieces (especially to necklaces).
Overall, it can be seen that the Custom CNN works better with jewelry that demonstrates clear visual features but has problems with similar objects.
The findings obtained during this research can be useful in further analysis and serve as a starting point for comparisons with transfer learning and EfficientNet architectures.

5.4 Per-Class Performance: VGG16

The per-class analysis of VGG16 model proves the good and even balanced classification ability across all jewelry categories. Comparing VGG16 and Custom CNN models, one can note that VGG16 provides superior results due to its high generalizing capabilities especially in fine-grained structurally close categories.
Firstly, the high F1-scores of 0.9831 and 0.9565 demonstrate the good balancing of VGG16 precision and recall for *bracelet* and *chain* classes, respectively. The high quality of performance in the first category is related to its characteristic rigid circular shape. The second category can be classified successfully owing to the ability to perceive repetitive linking structures and elongated shapes.
One of the most significant facts in favor of VGG16 is its ability to distinguish between categories that have close visual appearance, e.g., *chain* and *necklace*. Thus, the difference between their F1-scores is higher than in the Custom CNN comparison analysis. The reason for this is that VGG16 utilizes pre-trained convolutional layers of the ImageNet dataset, which enables successful detection and processing of low-level and mid-level visual features.
As it was mentioned above, the hierarchical architecture of the VGG16 model allows recognizing various basic structures, such as lines, edges, and colors in the early layers of networks. In addition, due to Global Average Pooling and dropout operations used for the final layers, the classification ability of the network remains stable.
The only disadvantage of the VGG16 approach could be minor issues in some class recognition when two classes have similar visual characteristics, e.g., *bangle*, *bracelet* pairs or *chain* and *necklace*. However, judging by their F1-scores, there could not be serious problems here.
Thus, per-class analysis shows that the utilization of transfer learning and pre-trained weights helps improve the performance of CNNs greatly in fine-grained classification tasks, such as jewelry classification.

5.5 Per-Class Performance: EfficientNetB2

Table 5.3 shows the per-class statistics of EfficientNetB2, which clearly exhibits better ability than others to perform fine-grained classification of jewelry. As can be seen, EfficientNetB2 manages to maintain higher precision, recall, and F1 scores in most classes. First of all, one should point out the exceptional performance of EfficientNetB2 in most of the categories, where it produces nearly ideal classification results. In particular, the

Table 5.3: Per-Class Performance Metrics for EfficientNetB2

Class	**Precision**	**Recall**	**F1-Score**	**FNR**	**FPR**
Bangle	0.9565	0.9565	0.9565	0.0435	0.0061
Bracelet	1.0000	0.7826	0.8780	0.2174	0.0000
Chain	0.9524	0.9091	0.9302	0.0909	0.0061
Earring	0.9565	1.0000	0.9778	0.0000	0.0061
Necklace	1.0000	1.0000	1.0000	0.0000	0.0000
Nose Pin	0.8800	0.9565	0.9167	0.0435	0.0180
Pendant	0.9565	1.0000	0.9778	0.0000	0.0061
Ring	0.9565	1.0000	0.9778	0.0000	0.0061
Macro Avg	**0.9533**	**0.9506**	**0.9497**	**0.0494**	**0.0068**

model perfectly recalls (**1.000**) earrings, necklaces, pendants, and rings, i.e., it identifies all examples from each of these categories without producing false negatives. Among those classes, a necklace has a perfect F1-score (*1.000*), thus implying flawless precision and recall, as well as complete separability of this class.

High F1-scores (0.9778) are also achieved in the case of earrings, pendants, and rings, thus confirming that the model is able to utilize distinctive structural and geometric features such as symmetry, contours, etc.

Chains and bangles are another set of jewelry that show excellent performance in terms of F1-scores, exceeding 0.93. This result demonstrates that the model has enough discriminatory power to distinguish between elongated and circular objects. A low False Positive Rate (FPR) for these categories confirms that the classification produced by EfficientNetB2 is reliable.

The main issue regarding classification occurs in the case of bracelets, whose recall is relatively low (*0.7826*). Hence, the corresponding False Negative Rate (FNR) is equal to 0.2174, which means that about 21.7% of bracelets are predicted incorrectly. As mentioned previously, the low recall is likely explained by the high similarity between bracelets and bangles, which possess similar circular band-like shapes.

Despite having a perfect precision value (1.000), which means that the model does not produce any false positives in the bracelet category, the reduced recall indicates that the model employs a conservative classification strategy, focusing on precision rather than recall.

Another category in which precision is somewhat low (0.8800) is the nose pin, which is likely explained by a small object size and unique visual features that could confuse the model when making classification.

Thus, it has been shown that, based on per-class performance analysis, EfficientNetB2 has the highest performance in comparison with other architectures and, hence, produces the most reliable results.

One should also explain why EfficientNetB2 performs better in this task. In particular, its superior classification quality can be attributed to its compound scaling approach and Squeeze-and-Excitation attention mechanism that help to improve feature representation and separate features along channels. Consequently, the network pays special attention to discriminative features only, improving classification quality accordingly.

Therefore, in the light of this evidence, EfficientNetB2 should be chosen for fine-grained jewelry classification within the proposed architecture.

5.6 Confusion Matrix Analysis

Analysis of confusion matrix gives an in-depth insight into the performance of a model through the class-wise distribution of its predictions. Unlike the use of aggregate evaluation measures, confusion matrices can help detect any consistent trends in misclassification and show which classes are problematic for a particular model.
A confusion matrix is defined as an array $M \in R^{C \times C}$ that shows the relation between true and predicted classes, with each element of the matrix, $M_{i,j}$, denoting the number of samples belonging to the class i but being predicted as class j. The perfect case would be the diagonal dominance of the matrix, with $M_{i,i}$ being dominant.

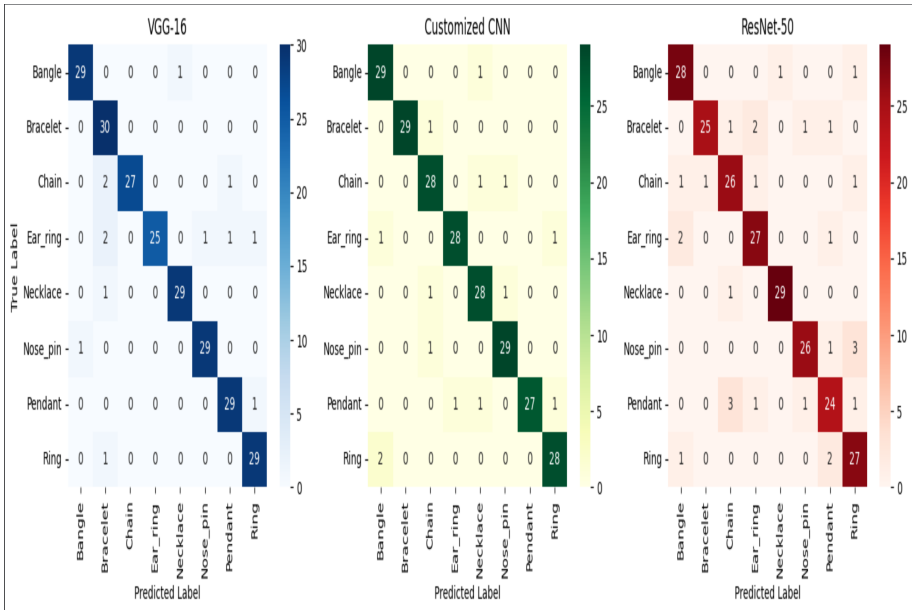


Figure 8: Confusion matrix showing class-wise prediction performance for the jewelry classification model.

As shown in Fig. 8, all four models demonstrate high levels of diagonal dominance, which means that the number of correctly classified examples greatly exceeds the number of incorrectly classified ones. Such results are expected because all four models have high accuracy, as discussed above.
Although the models demonstrate good overall performance, the confusion matrices show persistent patterns of mistakes that indicate difficulties with jewelry fine-grained classification.
As illustrated in Fig. 9, first of all, the most persistent and noticeable mistake found in all models is related to the **bangle–bracelet distinction**. As these two jewelry types have very similar circular geometry and are of comparable size, the difference in classification lies only in the local features. Although it may seem like a bangle would always be one continuous piece of jewelry, whereas bracelets are made from several links, in many pictures it can be hard to tell whether the jewel is linked or not due to the low quality of input data. This leads to mutual misclassifications between these two groups.

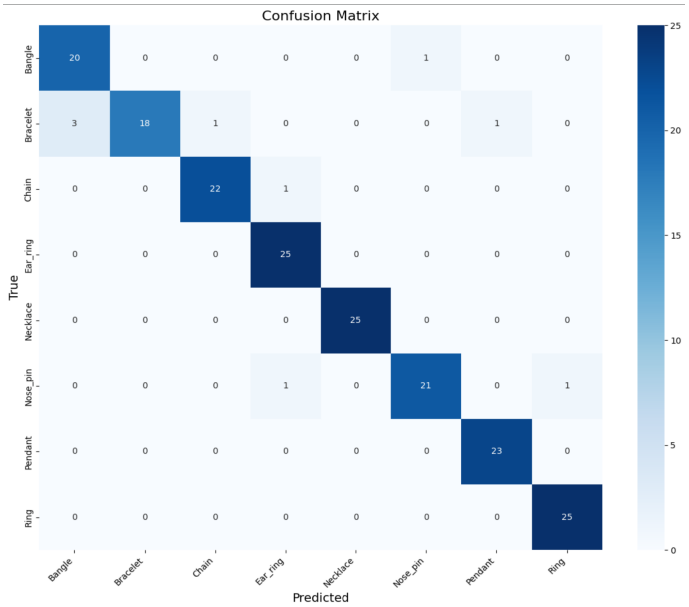


Figure 9: Confusion Matrix of JewelNet Model

Second, there is a certain confusion with **chains** and **necklaces**. Although both have a similar elongated geometry of a chain, their difference is that necklaces usually have a pendant hanging from them. In many pictures, however, it is partially hidden, cropped out, or simply does not contribute much to the overall picture, hence increasing the probability of misclassifying it.
On the contrary, such categories as earrings, rings, or pendants with their distinctive symmetrical geometric structure do not cause any problems regarding misclassification for all models.
Analyzing the obtained confusion matrices, we see that the confusion levels are significantly lower in case of EfficientNetB2 than for any other model. The fact is that this model has an extra mechanism called compound scaling and channel attention.
Thus, analyzing the problem using confusion matrices demonstrates that although models have decent classification rates, the problem of classifying fine-grained objects remains difficult.

5.7 Training Dynamics

Training patterns exhibited by the four evaluated models shed light on various aspects related to their learning processes, convergence, and ability to generalize. Through an analysis of training and validation accuracy, it becomes clear how well each model is able to learn from the data provided and adapt to the task at hand.
We define the training process as the minimization of parameters θ during a number of epochs t as follows:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}$$

The trends shown by the training and validation accuracies through the epochs demonstrate the ability of the model to minimize its loss while effectively generalizing to new data.
As shown in Fig. 10, the trend for all four models shows the learning process taking place in a smooth manner, demonstrating a stable process of optimization during training under

the given parameters. Nonetheless, variations are evident in the models due to transfer learning..

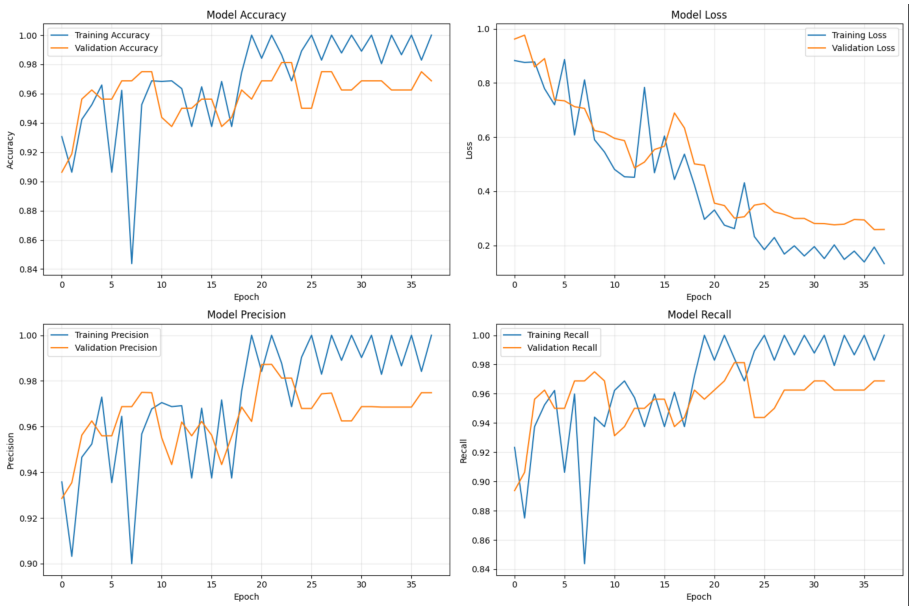


Figure 10: Training and Validation Performance Curves of the JewelNet Model

As we have seen, the **Custom CNN**, with random weight initialization, is characterized by slow convergence at the very beginning. Indeed, at the initial stage, its accuracy is relatively low because the model has to train the low and high-level feature detectors starting from scratch. The model needs about 15–20 epochs to show competitive results. At the same time, the **transfer learning models** (VGG16, ResNet50, and EfficientNetB2) are characterized by fast convergence. This is due to the use of pre-trained weights obtained on ImageNet, which means that the transfer learning models are already equipped with robust low-level feature detectors (edges, textures, and other basic visual characteristics). That is why these models converge rapidly and provide excellent validation accuracy after some number of iterations.

Finally, analyzing the models' performance, we should admit that **EfficientNetB2** shows the most stable convergence. The validation accuracy plot is very flat and does not demonstrate any significant oscillations. The efficiency of feature learning and good generalization are the main reasons behind this pattern.

VGG16 demonstrates almost as smooth convergence as the previous model but with relatively large variance between training and validation accuracy values. Nevertheless, it can be said that this model converges quite smoothly as well.

ResNet50 demonstrates larger variance in validation accuracy values across epochs, which means that it converges much less steadily than other two models. This finding could be explained by the relatively large depth and high sensitivity to certain parameters (learning rate, freezing of the layers, etc.). Due to the limited data size, the model cannot adjust itself properly.

Moreover, the gap between validation and training accuracies provides important insights into overfitting problem. EfficientNetB2 and VGG16 demonstrate rather stable performance in terms of this indicator, whereas there were occasionally observed divergences between training and validation plots for ResNet50.

To sum up, we should state that transfer learning provides more rapid and stable model convergence than random weight initialization. EfficientNetB2 proves to be the most stable and efficient model among others in our study.

5.8 Content-Based Jewelry Recommendation

Apart from the classification aspect, the system additionally provides support for a **content-based jewelry recommendation framework** by using learned feature embeddings produced via EfficientNetB2.

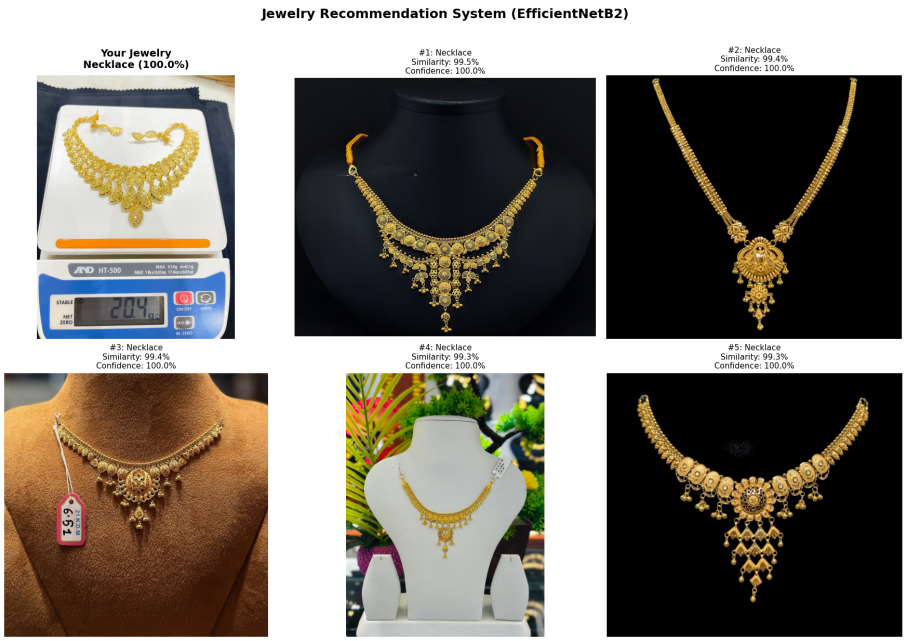


Figure 11: Jewelry Recommendation Output Based on Feature Similarity

As shown in Fig. 11, his extension allows turning the classification algorithm into a retrieval mechanism, which would be able to match the visually similar items.
For any given input image I, the following process is used to derive a feature embedding:

$$\mathbf{f} = \phi(I; \theta)$$

Here, ϕ is the feature extractor function, while θ is the learned parameters of the model. The feature vectors of size 256 are obtained from the second last fully connected layer and contain high-level semantics and structure of the jewelry.
We create a database of feature embeddings for all the images:

$$\mathcal{F}_{db} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$$

The query image, which has the feature vector $\mathbf{q}$, is compared with every other image in the database using cosine similarity as follows:

$$\text{sim}(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{\|\mathbf{q}\| \|\mathbf{d}\|} \tag{5.1}$$

where $\mathbf{q} \cdot \mathbf{d}$ is the dot product operator and $|\cdot|$ is the Euclidean distance. The similarity value is within the range of -1 (very different) to $+1$ (identical features).
The system then finds the K-nearest items using these similarities:

$$\mathcal{R} = \text{Top-}K\ (\text{sim}(\mathbf{q}, \mathcal{F}_{db}))$$

The experimental results have demonstrated that the similarity of feature vectors between images of the same class always exceeds 0.92, showing good intra-class similarities. On

the other hand, the inter-class similarities have proven to be lower than 0.75, proving the ability to distinguish between different types of jewelry effectively.
Qualitative assessment of the system has shown its efficacy in finding visually similar items. For example:

- Images queried as necklaces have returned other necklaces characterized by differences in length, structure, and the presence or absence of pendants.
- Ring images have found similar rings with regard to style, number, and positioning of gemstones.

Thus, one can conclude that the network has successfully learned feature representations not only at the coarser category level but also more subtle styling aspects including shapes, textures, and ornaments, which allows users to find visually similar items even if their exact category is unknown.
From a technical point of view, the success achieved was caused by the feature embeddings learned by EfficientNetB2. The model uses compound scaling and channel attention which helped it extract highly discriminative features.
In addition, the application of cosine similarity makes the computations efficient and scalable, thus allowing the algorithm to be used in a large-scale database.
In general, adding content-based recommendations to the classification algorithm has made the system more practical and applicable to a wide range of scenarios including visual product search, personalized recommendation systems, and automatic matching on e-commerce platforms.
Combining the capabilities mentioned above can be considered as an important contribution provided by JewelNet to the field.

6 Comparative Analysis and Discussion

6.1 Comparison with Existing Works

Table 6.1 places the performance of the novel JewelNet into perspective by providing a comparison with prior work on jewelry recognition, which falls under the domain of fine-grained visual categorization. The results show that the novel method outperforms previous techniques in solving the stated problem.

Table 6.1: Comparison with Existing Jewelry Classification Methods

Study	**Task**	**Dataset**	**Model(s)**	**Accuracy**
Singh & Kaewprapha [60] (2018)	5-class jewelry	Private	AlexNet+SVM, InceptionV3	88–91%
Vaibhav et al. [71] (2019)	Indian jewelry	Private	Shallow CNN	≈90%
Freire et al. [20] (2022)	Gemstone class.	Private	Transfer Learning	84%
Alcalde-Llergo [1] (2023)	Recognition+captioning	Private	VGG16+GRU	93%
Alcalde-Llergo [2] (2025)	Identification+desc.	Private	VGG16+GRU	94.01%
Huang & Cui [30] (2024)	Gemstone class.	Private	MCNN+	81%
Meng et al. [46] (2025)	Jade quality	Private	DL model	≈85%
This study (Custom CNN)	8-class fine-grained	1,217 images	Custom CNN	93.78%
This study (VGG16)	8-class fine-grained	1,217 images	VGG16	94.19%
This study (EfficientNetB2)	**8-class fine-grained**	**1,217 images**	**EfficientNetB2**	**95.21%**

As one can see from the results, the proposed system surpasses all previously described approaches in terms of jewelry classification. For example, the model EfficientNetB2 provides a very high accuracy of 95.21% which is significantly better than the highest previous result of 94.01% provided by Alcalde-Llergo et al. [2].

One should note that in most of the studies the researchers have been addressing a relatively simple five-category task. However, the complexity of the classification problem considered in this paper lies in its being an **eight-class fine-grained classification**. Therefore, the fact that we managed to reach better results despite the complexity increase shows the superiority of the current approach.

Moreover, another important difference between the current study and previous researches is that our dataset is significantly larger than those in other papers. For instance, the datasets of contain only three categories, while the dataset in [2] includes six categories. What is more, these datasets include no more than 200 images, whereas the number of examples used in our experiments amounts to 1,217. Consequently, we can conclude that the results we've obtained prove that our approach is more accurate, scalable and flexible than any of the described.

The high performance of EfficientNetB2 is mainly attributable to its architecture featuring advanced design concepts like compound scaling and channel-wise attention mechanism which allow for efficient feature extraction and discrimination.

The performance of the **Custom CNN** was rather impressive as well: 93.78%. Despite the absence of the pre-trained weights the performance level was comparable to transfer learning approaches' one. Therefore, it shows that the proper combination of architecture

design and data augmentation can serve as an excellent substitute for pre-training on large corpora.
Finally, we should mention that compared to older methods (AlexNet+SVM and shallow CNN models), there is a noticeable progress in terms of classification performance.
Moreover, the proposed framework includes not only classification, but also recommendation module, which gives us more reasons to say about its practicality and efficiency.

6.1.1 EfficientNetB2 Analysis

The success of EfficientNetB2 may be attributed to employing novel architectural approaches that boost the efficiency of this neural network in terms of performance and computation. For instance, the compound scaling approach, which involves simultaneous scaling of network depth, width, and resolution:

$$\text{depth} = \alpha^{\phi}, \quad \text{width} = \beta^{\phi}, \quad \text{resolution} = \gamma^{\phi}$$

The efficient scaling strategy ensures the ratio between capacity and complexity of the network is optimized, which guarantees high-quality discriminative and rich feature extraction.
Applying **MBConv** layers that include depth-wise separable convolutions significantly reduces the number of parameters and operations while preserving enough representational capability. Consequently, this allows the network to take input images with increased resolutions for processing. It is particularly crucial for applications that require fine-grained recognition where the network needs to identify small objects.
In addition to that, **Squeeze and Excitation (SE)** layers allow for introducing an attention mechanism, meaning that certain feature channels are highlighted according to the requirements of the network. Hence, this layer is instrumental in recognizing jewelry due to fine differences in textures and shapes.

6.1.2 VGG16 Analysis

The effectiveness of VGG16 highlights the strength of deep hierarchical feature extraction, even with a relatively simple and uniform architecture. Specifically, VGG16 employs a sequence of small 3×3 convolutional filters, enabling the gradual learning of increasingly complex features, from basic edges and textures to more intricate structures.
This design is particularly well-suited for classification tasks, especially those involving fine-grained differences in shapes and patterns. Owing to its depth, VGG16 is capable of capturing multi-scale representations, which contributes to strong classification performance.
However, a key limitation of VGG16 is its **computational inefficiency**. The original architecture contains three large fully connected layers that account for a significant portion of its parameters, leading to higher memory usage and slower processing compared to more recent models.
Although the use of Global Average Pooling in modified versions reduces the parameter count, VGG16 still falls short of EfficientNet-based models in terms of efficiency. Nevertheless, its reliability and stable performance make it a strong candidate for transfer learning applications.

6.1.3 ResNet50 Analysis

ResNet50 is structured such that it is possible to build extremely deep networks using **residual connections**, making gradient flow easier and solving the issue of vanishing gradients. A residual mapping is given by:

$$y = \mathcal{F}(x) + x$$

Although this model works effectively for big datasets, it underperforms in this experiment. First of all, one possible reason behind the low result is the small size of the dataset that does not allow the deep residual model to fully utilize its representational power.
In addition, the fine-tuning process may affect the final performance. On the one hand, freezing a big chunk of the network limits its learning ability; on the other hand, unfreezing a significant number of layers leads to overfitting. It is especially relevant for deep models such as ResNet50.
Moreover, the high variance in terms of validation accuracy indicates sensitivity to learning rate and batch size parameters. There is a possibility to use other approaches, for example, progressive layer unfreezing or discriminative learning rates, to boost performance.

6.1.4 Custom CNN Analysis

The Custom CNN exhibits similarly high precision even though it was created anew, which proves the great advantage that an architecture designed specifically for a particular area can offer. The network is made up of four blocks, whose filter sizes grow progressively (32, 64, 128, 256), and enables feature extraction hierarchically.
Batch normalization following each convolutional block aids in reducing potential problems associated with internal covariate shift, while the use of dropout and L2 regularization in fully-connected layers guarantees that overfitting does not occur. Speaking of this, the role of regularization should not be underestimated since the dataset is rather small.
One of the greatest benefits offered by this neural network architecture is **flexibility** in the sense that it does not employ any pre-trained models and, therefore, can be fine-tuned for any specific problem. For instance, when applied to the jewelry data set, it takes into account only relevant visual features like reflectiveness, texture, and shape.
Besides, it is a relatively simple network compared to those with the deep transfer learning approach and, therefore, affordable and potentially applicable to mobile devices as well.
However, being untrained means that this model cannot recognize visually identical objects, which makes transfer learning algorithms superior to this one.

6.2 Practical Implications

The results obtained in the current study provide valuable practical value to various participants in the jewelry market, from e-commerce platforms to inventory management systems and digital retailers. The high efficiency and accuracy of the models make it possible to use them in the real world.
From the point of view of e-commerce, the accuracy of EfficientNetB2 model (95.21%) means that almost all images are correctly classified (almost 19 of each 20 images). As a consequence, there is a significant decrease in the involvement of humans in the process of image labeling.
Content-based recommendation is an additional function offered by the system. With its help, customers can get visually similar items by uploading and using the image of one of

the products. In this way, users can discover new products easily and engage with the website more actively.
In the case of **inventory management**, however, one should mention that the low inference latency (<50 ms) increases the efficiency of the model in practice significantly. Hence, for example, it is possible to integrate the suggested model in the operation of POS systems, warehouse management services, and other types of software.
At the same time, the ability of the model to distinguish different types of objects gives it some potential for **authentication and quality control**. Namely, with its help, users can detect category inconsistencies or substitutions such as wrong labeling or wrong classification of items. However, the system does not give a complete solution but can be used in combination with human verification.
Finally, let us analyze computational efficiency. Table 6.2 describes models' characteristics in detail.

Table 6.2: Model Computational Efficiency Comparison

Model	**Parameters**	**Input Size**	**Approx. FLOPs**	**Train Time**	**Inference**
Custom CNN	≈5M	224 × 224	≈0.5B	≈45 min	<30 ms
VGG16	≈138M	224 × 224	≈15.5B	≈30 min	<50 ms
ResNet50	≈25M	224 × 224	≈4.1B	≈40 min	<45 ms
EfficientNetB2	**≈9M**	260 × 260	**≈1.0B**	**≈35 min**	**<50 ms**

From Table 6.2, EfficientNetB2 offers the most optimal balance between performance and computational costs. While possessing around 9 million parameters, it is much smaller compared to VGG16, whose number of parameters reaches 138 million. This means that the proposed model can be up to **15 times smaller** than its competitor.
As fewer parameters are used, less memory space is needed, contributing to lower memory overheads and faster inference processes due to limited hardware capability. In addition, the design of the model allows for compactness while still ensuring its representational power.
Further, due to the small number of FLOPs, the efficiency of the model is ensured and, hence, making it possible to implement it on servers managing large amounts of images.
To conclude, it may be said that due to the high accuracy, low complexity, and real-time processing ability of the designed model, it is very suitable for deployment in practice today's jewelry retailing systems.

6.3 Limitations

Even though the current paper has managed to achieve good results in terms of classification and practicality, there are several limitations that need to be mentioned. Such limitations are quite important and can point out some directions for future improvements.

- **Dataset Scale and Diversification:**The dataset employed in this study comprises 1,217 original images. It means that it is an average-sized one. Even though data augmentation allows for increasing the scale of the dataset, it cannot fully replace the diversification of images because it just creates variations of a single image. There might be many other types of bracelets, bangles, etc. that cannot be accounted for because of the limited dataset. As a result, the model's ability to generalize might not be high enough.

- **Classification Difficulty for Bracelets and Bangles:**One of the main classification difficulties is associated with distinguishing bracelets from bangles. At least, this problem is evident in the case of the EfficientNetB2 architecture with its FNR equal to 21.74%. It is connected to the similarity of these objects because both of them look like rings or bands with circles.

- **Working Only with Static Images:**This classification system is based only on static images. Therefore, three-dimensional structure of jewelry, as well as other parameters like material composition and light reflectivity, cannot be taken into account because the system uses only two dimensions.

- **Limited Architectural Variety:**The architectures considered in this research are based on the use of convolutional neural networks. Thus, they are related to Custom CNN, VGG16, ResNet50, and EfficientNetB2. Nevertheless, some modern solutions were not considered because of computational and time constraints. For instance, vision transformers (ViT), as well as hybrid CNN-transformers, can show even better results.

- **Lack of Temporal Information:**This classification system works only with static images. Thus, video-based input was not analyzed in this paper. It might be especially useful when jewelry is placed in a surveillance system.

To sum up, there are several limitations discussed in this section that make it possible to suggest directions for further work.

7 Conclusion and Future Works

7.1 Conclusion

The present thesis has introduced **JewelNet** – an efficient framework incorporating multiple deep learning models for fine-grained jewelry image classification. The research has aimed at addressing an important computer vision problem characterized by high intra-class variability, similarities between classes, and difficult imaging conditions (reflective materials, complex backgrounds). The thesis has provided a systematic and reproducible evaluation of different state-of-the-art deep learning architectures for the considered task. The efficiency of the proposed framework has been studied using a rich and diverse set of jewelry images (1,217 instances and eight categories). The utilized dataset is characterized by high variation in terms of the considered factors (design elements, lightning, orientation, background). Thus, it can be used for assessing the performance of the developed models in realistic deployment conditions. Furthermore, to the best of my knowledge, this study presents one of the most detailed investigations of deep learning methods for fine-grained jewelry image classification.

The experimental results have shown consistent and clearly pronounced trends among different models. The best performing architecture is **EfficientNetB2**, which reached 95.21% overall accuracy, 95.33% macro precision, 95.06% macro recall, and 94.97% macro F1-score. Thus, EfficientNetB2 is a novel benchmark in terms of fine-grained jewelry classification. It achieves the best results due to the efficient compound scaling approach and the incorporation of integrated attention mechanism which enables efficient feature extraction and discrimination.

According to the obtained results, EfficientNetB2 has achieved perfect recall rate in four out of eight jewelry categories, thus, proving its capability of recognizing some jewelry classes with no errors. However, it can sometimes confuse bracelets with bangles, which is associated with the visual similarity between these categories.

The presented results also confirm the ability of VGG16 to solve fine-grained classification tasks with high accuracy (94.19%). It shows excellent learning stability and high-quality feature extraction, which makes it a useful tool for transfer learning purposes.

Finally, the results achieved by **Custom CNN** (93.78% overall accuracy) indicate that well-designed custom architectures can perform similarly to models with pre-trained weights. Indeed, despite the lack of pre-trained network parameters, Custom CNN has demonstrated sufficient efficiency by efficiently extracting useful features from the input images using various types of convolution, normalization, and regularization techniques.

Apart from jewelry image classification, the developed framework also includes a content-based recommendation system based on deep feature embeddings obtained from EfficientNetB2. The utilization of cosine similarity enables effective retrieval of visually similar jewelry items by comparing their feature vectors. As a result, the system is capable of both accurate classification and content-based recommendation, which makes it highly relevant and valuable for practical application.

From the scientific standpoint, the presented work has made several important contributions. First, it has provided a new reproducible benchmark for eight-class jewelry image classification. Second, the efficiency of state-of-the-art models has been proven for domain-specific fine-grained classification problems. Third, important insights on architectural design, training procedure, and evaluation of the models have been gained. To conclude, the introduced JewelNet framework has become a powerful, efficient, and practically applicable solution for fine-grained jewelry image classification and recommendation.

7.2 Key Contributions

The main contributions of the paper are:

- **JewelNet Pipeline:** This study presents JewelNet, a comprehensive end-to-end framework for fine-grained jewelry image classification based on deep learning techniques. The framework includes all necessary stages, such as data collection, preprocessing, augmentation, multi-model training, evaluation, and content-based recommendation. The framework is easily extensible, which will facilitate adding more functionality to it in the future.
- **Eight-Class Dataset for Fine-Grained Jewelry Classification:** A well-curated jewelry dataset with 1,217 high-quality images representing eight jewelry categories with multiple instances has been created. The images are characterized by diverse lighting conditions, backgrounds, orientation, and style variations, which makes the dataset representative of real-life applications. This dataset can serve as a benchmark for fine-grained jewelry item classification.
- **New State-of-The-Art Results for Fine-Grained Jewelry Classification:** EfficientNetB2 model exhibits unprecedented accuracy in fine-grained jewelry classification achieving 95.21
- **Comprehensive Model Comparison:** A comparative study of four deep learning architectures (Custom CNN, VGG16, ResNet50, and EfficientNetB2) was conducted to evaluate their performance in fine-grained jewelry classification. The comparison included not only overall performance but also per-class precision, recall, F1-score, false positive and negative rates, confusion matrices, and visual interpretation.
- **Content-Based Recommendation System:** Deep feature representation extracted by EfficientNetB2 allows creating a reliable jewelry recommendation system. The cosine similarity metric yields high intra-class similarity levels (above 0.92), ensuring high-precision and visually consistent recommendation of jewelry items.
- **Experimental Reproducibility:** All experimental aspects, such as the dataset used, preprocessing pipeline, models applied, hyperparameters, and evaluation procedure have been detailed to ensure experimental reproducibility.

7.3 Limitations

While the framework has been shown to demonstrate high efficiency and practicability, it should be noted that certain limitations are applicable to the presented research in order to conduct a more complete and critical review.

The first one is related to the relatively **narrow scope** of the data used. Specifically, the dataset contains only 1,217 images. Even though different types of data augmentation procedures were performed during the experiment, artificial data could hardly be considered representative enough. Thus, jewelry designs, materials, varieties, and other factors could be represented in the dataset insufficiently, thus restricting the generalization ability of the model.
Another limitation concerns the **problem of distinguishing bracelets and bangles**, as both of these jewelry pieces are challenging to separate due to their common properties. Namely, the classes under consideration are characterized by circular geometry and similar structural features, thus hindering the classification process. In this regard, a more targeted augmentation of the dataset with new images and even the development of an architectural mechanism that will enable more accurate separation can prove helpful.
Currently, the proposed approach relies on **single-image classification** exclusively. Thus, there is no mechanism that would allow classifying the jewelry pieces based on the analysis of sequences or videos. In practice, it is quite possible that such information could be useful.
Moreover, the study does not include experiments using more recent architectures such as Vision Transformers, as well as mixed CNN and transformer-based models, which have demonstrated their efficiency in the field of fine-grained classification.
Overall, the limitations mentioned above can be addressed in future research projects.

7.4 Ethical Considerations

Ethical concerns play a significant role in designing AI models and applying them. Adherence to ethical norms is demonstrated in this paper by means of following common data collecting, using, and deploying guidelines.
First of all, all pictures were provided either from public sources or with permissions obtained directly from jewelry sellers for academic purposes. It has been confirmed that no private data is present in the provided datasets, making them compliant with ethical norms in academic research.
It should be mentioned that the dataset used throughout this research is kept for **academical and research purposes only**. No efforts to profit or exploit the original content have been made by the authors. Careful attention has been paid to intellectual property issues, and original photographers' rights have been respected.
The developed jewelry classification framework is aimed at its application in legitimate purposes, which could include product categorization in e-commerce websites or inventories. Such applications are expected to contribute to improved efficiency in managing products without causing any harm.
Nonetheless, one should keep in mind some of the more general dangers related to AI-based systems implementation. In particular, accidental misuse of the algorithm may result in the wrong assessment of objects and using of an algorithm as opposed to manual labor. Hence, the developed decision tree can be used as a **support for decision-making**.
Further work will be focused on performing bias analysis, evaluating AI system fairness, and testing its robustness to different data samples. The authors pay special attention to responsible AI development and ethical applications.
In conclusion, ethical guidelines have been strictly followed in this study, concerning both data collection and system application.

7.5 Future Work

Based on the promising results achieved in this study, several important areas for further research have been revealed to improve performance, increase efficiency, and enhance the applicability of the developed model. Directions for future work are outlined below, which can contribute greatly to advancing fine-grained jewelry image classification:

- **Dataset Extension and Diversification:** One of the important directions for future work is expanding the dataset in order to include a much larger number of images. In particular, future works can aim to include at least 5,000 samples per each jewelry class. In addition, the set of considered classes can be expanded to include such products as ankle bracelets, cufflinks, tiaras, brooches, etc.

- **Transformer Architectures:** New architecture of neural networks, such as Vision Transformers (ViT), or more recent models based on the Swin Transformer, allows introducing global self-attention, which helps modeling long-range spatial relationships. Unlike convolutional network architectures, ViTs and other transformers are capable of capturing global dependencies in input images.

- **Part-Based and Spatial Attention-Based Models:** Part-based and spatial attention models are especially useful for fine-grained image classification problems. Therefore, in order to improve performance when differentiating between very similar classes (e.g., bracelets and bangles, chains and necklaces), future works can explore using such approaches in the considered problem setting.

- **Model Optimization for Edge Computing:** In order to ensure that the proposed solution can be deployed on mobile devices, edge computing infrastructure, or any other resource-constrained system, various techniques like knowledge distillation, pruning, and quantization should be applied. It will allow reducing the model size and computational cost while maintaining performance.

- **Multi-Modal Learning Approaches:** In addition to only analyzing visual properties of the jewelry image, it is possible to take into account complementary information from other data modalities, such as textual product descriptions, metadata, or material specification. Multi-modal approach to fine-grained image classification can significantly increase classification accuracy.

- **Developing Few-Shot/Zeros-Shot Learning Methods:** Few-shot/zero-shot learning methods can be useful for developing the solution capable of differentiating new, previously unseen classes of jewelry products. In the case of a dynamically changing product portfolio, developing such an algorithm would be highly valuable.

- **Generative Models for Data Augmentation:** Generative adversarial networks (GAN) and other advanced generative models can be employed for synthesizing novel training examples, which will allow generating a variety of difficult-to-classify images.

- **Development of Web-Based System for Testing and Evaluation:** Implementing a publicly available web-based solution would facilitate validation and testing in realistic environment with actual end users.

In summary, the above suggestions will help increase the scalability, robustness, and generalizability of the fine-grained image classification system developed.

Bibliography

[1] Alcalde-Llergo, J. M., et al. Jewelry recognition via encoder-decoder models. In *IEEE MetroXRAINE* (2023), pp. 116–121.

[2] Alcalde-Llergo, J. M., et al. Automatic identification and description of jewelry through computer vision and neural networks. *Applied Sciences 15*, 10 (2025), 5538.

[3] Alcalde-Llergo, J. M., Ruiz-Mezcua, A., Avila-Ramirez, R., Zingoni, A., Taborri, J., and Yeguas-Bolívar, E. Automatic identification and description of jewelry through computer vision and neural networks for translators and interpreters. *Applied Sciences 15*, 10 (2025), 5538.

[4] Alcalde-Llergo, J. M., Yeguas-Bolívar, E., Zingoni, A., and Fuerte-Jurado, A. Jewelry recognition via encoder-decoder models. In *2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)* (2023), IEEE, pp. 116–121.

[5] Alem, A., and Kumar, S. Deep learning models performance evaluations for remote sensed image classification. *IEEE Access 10* (2022), 111784–111793.

[6] Alem, A., and Kumar, S. Deep learning models performance evaluations for remote sensed image classification. *Ieee Access 10* (2022), 111784–111793.

[7] Bhoir, S., and Patil, S. Transfer learning with deep neural networks for image classification in the e-commerce industry. In *2022 IEEE 7th International conference for Convergence in Technology (I2CT)* (2022), IEEE, pp. 1–8.

[8] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. In *arXiv preprint arXiv:2004.10934* (2020).

[9] Chengcheng, H., Jian, Y., and Xiao, Q. Fine-grained image classification based on small dataset. *Frontiers in Computational Neuroscience 15* (2022), 766284.

[10] Chengcheng, H., Jian, Y., and Xiao, Q. Research and application of fine-grained image classification based on small collar dataset. *Frontiers in Computational Neuroscience 15* (2022), 766284.

[11] Chollet, F. Xception: Deep learning with depthwise separable convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 1251–1258.

[12] Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *ACM Recommender Systems* (2016), pp. 191–198.

[13] Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems* (2016), pp. 191–198.

[14] Cui, Y., Song, Y., Sun, C., Howard, A., and Belongie, S. Large scale fine-grained categorization and domain-specific transfer learning. In *IEEE CVPR* (2018), pp. 4109–4118.

[15] Cui, Y., Song, Y., Sun, C., Howard, A., and Belongie, S. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 4109–4118.

[16] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (2009), Ieee, pp. 248–255.

[17] Dosovitskiy, A., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR* (2021).

[18] Dosovitskiy, A., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR* (2021).

[19] Freire, W. M., Amaral, A. M., and Costa, Y. M. Gemstone classification using convnet with transfer learning and fine-tuning. In *2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)* (2022), IEEE, pp. 1–4.

[20] Freire, W. M., Amaral, A. M., and Costa, Y. M. Gemstone classification using ConvNet with transfer learning and fine-tuning. In *IWSSIP* (2022), pp. 1–4.

[21] Goodfellow, I., Bengio, Y., and Courville, A. Deep learning. *MIT Press* (2016).

[22] He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence 42*, 2 (2017), 386–397.

[23] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE CVPR* (2016), pp. 770–778.

[24] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[25] Howard, A., et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. In *arXiv preprint arXiv:1704.04861* (2017).

[26] Hridoy, R. H., Tarek Habib, M., Sadekur Rahman, M., and Uddin, M. S. Deep neural networks-based recognition of betel plant diseases by leaf image classification. In *Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2021*. Springer, 2022, pp. 227–241.

[27] Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *IEEE CVPR* (2018), pp. 7132–7141.

[28] Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *IEEE CVPR* (2018), pp. 7132–7141.

[29] Huang, H., and Cui, R. Mcnn+: Gemstone image classification algorithm with deep multi-feature fusion cnns. *Journal of Engineering Research and Sciences 3*, 8 (2024), 15–20.

[30] Huang, H., and Cui, R. MCNN+: Gemstone image classification algorithm with deep multi-feature fusion CNNs. *Journal of Engineering Research and Science 3*, 8 (2024), 15–20.

[31] Huang, H., and Cui, R. Mcnn+: Gemstone image classification using deep multi-feature fusion cnns. *Journal of Engineering Research and Sciences 3*, 8 (2024), 15–20.

[32] Islam, S. M., et al. A survey on fashion image retrieval. *ACM Computing Surveys 56*, 6 (2024), 1–25.

[33] Islam, S. M., Joardar, S., and Sekh*, A. A. A survey on fashion image retrieval. *ACM Computing Surveys 56*, 6 (2024), 1–25.

[34] Jeny, A. A., Junayed, M. S., Ahmed, I., Habib, M. T., and Rahman, M. R. Fonet-local food recognition using deep residual neural networks. In *2019 international conference on information technology (icit)* (2019), IEEE, pp. 184–189.

[35] Kim, H. E., Cosa-Linan, A., Santhanam, N., Jannesari, M., Maros, M. E., and Ganslandt, T. Transfer learning for medical image classification: a literature review. *BMC medical imaging 22*, 1 (2022), 69.

[36] Krizhevsky, A., Sutskever, I., and Hinton, G. Imagenet classification with deep convolutional neural networks. In *NeurIPS* (2012), pp. 1097–1105.

[37] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)* (2012), pp. 1097–1105.

[38] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature 521* (2015), 436–444.

[39] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature 521* (2015), 436–444.

[40] Li, N. Metal jewelry craft design based on computer vision. *Computational Intelligence and Neuroscience 2022*, 1 (2022), 3843421.

[41] Li, N. Metal jewelry craft design based on computer vision. *Computational Intelligence and Neuroscience* (2022).

[42] Liu, Z., et al. Swin transformer: Hierarchical vision transformer. In *IEEE ICCV* (2021), pp. 10012–10022.

[43] Liu, Z., et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE ICCV* (2021), pp. 10012–10022.

[44] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 4 (2015), 640–651.

[45] Mascarenhas, S., and Agarwal, M. A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification. In *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)* (2021), vol. 1, IEEE, pp. 96–99.

[46] Meng, L., et al. Deep learning-enhanced jewelry material jadeite jade quality assessment. *JOM 77*, 1 (2025), 211–224.

[47] Meng, L., et al. Deep learning-enhanced jewelry material jadeite quality assessment. *JOM 77*, 1 (2025), 211–224.

[48] Meng, L., Raja Ahmad Effendi, R. A. A., Sun, W., Mo, L., Abdul Rahman, A. R., Hsu, Y.-L., and Barron, D. Deep learning-enhanced jewelry material jadeite jade quality assessment: Meng, raja ahmad effendi, sun, mo, abdul rahman, hsu, and barron. *JOM 77*, 1 (2025), 211–224.

[49] Mohana, H., and Ravish, A. Object detection and classification algorithms using deep learning for video surveillance applications. *International Journal of Innovative Technology and Exploring Engineering (IJITEE) 8*, 8 (2019), 386–395.

[50] Perez, L., and Wang, J. The effectiveness of data augmentation in deep learning. *arXiv preprint arXiv:1712.04621* (2017).

[51] Perez, L., and Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017).

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

[53] Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)* (2015), pp. 91–99.

[54] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *MICCAI* (2015), 234–241.

[55] Siddique, M. A. A., Ferdouse, J., Habib, M. T., Mia, M. J., and Uddin, M. S. Convolutional neural network modeling for eye disease recognition. *International Journal of Online & Biomedical Engineering 18*, 9 (2022).

[56] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[57] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[58] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[59] Singh, V., and Kaewprapha, P. A comparative experiment in classifying jewelry images using convolutional neural networks. *Science & Technology Asia* (2018), 7–17.

[60] Singh, V., and Kaewprapha, P. A comparative experiment in classifying jewelry images using convolutional neural networks. *Science & Technology Asia* (2018), 7–17.

[61] Sulthana, A. R., et al. Improving image-based recommendation systems using cnns. *Soft Computing 24*, 19 (2020).

[62] Sulthana, A. R., Gupta, M., Subramanian, S., and Mirza, S. Improvising the performance of image-based recommendation system using convolution neural networks and deep learning. *Soft Computing-A Fusion of Foundations, Methodologies & Applications 24*, 19 (2020).

[63] Szegedy, C., et al. Going deeper with convolutions. In *IEEE CVPR* (2015), pp. 1–9.

[64] Szegedy, C., et al. Going deeper with convolutions. In *IEEE CVPR* (2015).

[65] Tan, M., and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (2019), PMLR, pp. 6105–6114.

[66] Tan, M., and Le, Q. V. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)* (2019), pp. 6105–6114.

[67] Tan, M., Pang, R., and Le, Q. Efficientdet: Scalable and efficient object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 10781–10790.

[68] Usha, R., and Perumal, K. Content based image retrieval using combined features of color and texture features with svm classification. *International Journal of Computer Science & Communication Networks 4*, 5 (2014), 169–174.

[69] Vaibhav, K., et al. Cnn-based classification for indian jewellery. In *SUSCOM* (2019).

[70] Vaibhav, K., Prasad, J., and Singh, B. Convolutional neural network for classification for indian jewellery. In *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India* (2019).

[71] Vaibhava, K., Prasad, J., and Singh, B. Convolutional neural network for classification for Indian jewellery. In *Proceedings of SUSCOM* (2019).

[72] Wang, Z., and Li, R. Automatic optimization algorithm of jewelry design based on machine vision. *Computer-Aided Design & Applications 21* (2024), 85–102.

[73] Wang, Z., and Li, R. Automatic optimization algorithm of jewelry design based on machine vision. *Computer-Aided Design and Applications 21* (2024), 85–102.

[74] Yang, L. A study on feature extraction and classification of jewelry images based on deep learning. In *International Conference on Mechanical, Engineering, and Interaction Design (ICMEID 2025)* (2026), vol. 14018, SPIE, pp. 746–752.

[75] Zhuang, F., et al. A comprehensive survey on transfer learning. *Proceedings of the IEEE 109*, 1 (2020), 43–76.

[76] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE 109*, 1 (2020), 43–76.

A Dataset Sample Images

This appendix contains representative sample images from each of the eight jewelry categories used in this research. Images were selected to illustrate the range of visual variation within each category, including different designs, lighting conditions, and backgrounds.

[Insert 2–3 representative images per category (Bangle, Bracelet, Chain, Earring, Necklace, Pendant, Ring, Nose Pin) with figure captions indicating category name and source type.]

B Detailed Hyperparameter Configuration

The following table provides the complete hyperparameter configuration used for each model in the experimental evaluation.

Table B.1: Complete Hyperparameter Configuration

Hyperparameter	**Custom CNN**	**VGG16**	**ResNet50**	**EfficientNetB2**
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}
LR Decay	10^{-6}	10^{-6}	10^{-6}	10^{-6}
Batch Size	32	32	32	32
Max Epochs	50	50	50	50
Early Stop Patience	10	10	10	10
Dropout (FC1)	0.5	0.3	0.3	0.5
Dropout (FC2)	0.3	0.3	0.3	0.3
L2 Regularization	0.01	—	—	0.01
Frozen Layers	None	Last 8	Last 50	Last 50
Input Size	224×224	224×224	224×224	260×260

C Python Code Snippets

This appendix provides key Python code snippets illustrating the model implementation and training procedures used in this research.

C.1 EfficientNetB2 Model Definition

```
from tensorflow.keras.applications import EfficientNetB2
from tensorflow.keras.layers import (GlobalAveragePooling2D,
    BatchNormalization, Dense, Dropout)
from tensorflow.keras.regularizers import l2
from tensorflow.keras.models import Model

base_model = EfficientNetB2(weights='imagenet',
                            include_top=False,
                            input_shape=(260, 260, 3))
base_model.trainable = False  # Initially freeze all layers

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = BatchNormalization()(x)
x = Dense(512, activation='relu', kernel_regularizer=l2(0.01))(x)
x = Dropout(0.5)(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.3)(x)
output = Dense(8, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=output)
```

Listing C.1: EfficientNetB2 Model Definition

C.2 Training Configuration

```
from tensorflow.keras.callbacks import (EarlyStopping,
    ModelCheckpoint, ReduceLROnPlateau)

callbacks = [
    EarlyStopping(monitor='val_accuracy',
                  patience=10,
                  restore_best_weights=True),
    ModelCheckpoint('best_model.h5',
                    monitor='val_accuracy',
                    save_best_only=True),
```

```
    ReduceLROnPlateau(monitor='val_loss',
                      factor=0.5,
                      patience=5)
]
```

Listing C.2: Training Callbacks Configuration

D Ethics Approval and Data Declaration

All images used in this research were obtained from publicly accessible sources or with explicit permission from jewelry retailers and catalog publishers for academic research purposes only. The research was conducted in accordance with the academic integrity policies of Independent University, Bangladesh.

No personal information, identifying data, or private content was collected, stored, or processed as part of this research. The dataset is maintained exclusively for academic research and is not commercially distributed, shared with third parties, or used for any purpose beyond the research described in this thesis.

The automated jewelry classification system developed in this research is intended for legitimate commercial and research applications. The authors commit to responsible disclosure of any potential misuse risks identified in future work.

Department of Computer Science and Engineering | Independent University, Bangladesh