# ECE3296 Digital Image and Video Processing

# Content-Based Image Retrieval System Using Convolutional Neural Networks

# Assignment


Ali Karimeh
Faculty of Engineering
Multimedia University
Cyberjaya, Malaysia
1191302231@student.mmu.edu.my

Bashir Tawfig
Faculty of Engineering
Multimedia University
Cyberjaya, Malaysia
1181102921@student.mmu.edu.my

Yousef Yasser
Faculty of Engineering
Multimedia University
Cyberjaya, Malaysia
1171103943@student.mmu.edu.my

Marawan Eldeib
Faculty of Engineering
Multimedia University
Cyberjaya, Malaysia
1181102334@student.mmu.edu.my



***Abstract*—This paper introduces a sophisticated image classification system using convolutional neural networks for enhanced feature extraction and image retrieval. Focusing on noise reduction and image normalization, the system adeptly processes diverse image datasets. Results indicate high accuracy in classification and retrieval, underscoring the system's potential for varied automated image analysis applications.**



***Keywords—Image Classification, Convolutional Neural Networks, Feature Extraction, Image Preprocessing, Image Retrieval, Noise Reduction, Data Normalization.***


## I. INTRODUCTION

In the era of digital transformation, the ability to efficiently process and classify large volumes of image data is increasingly vital across various sectors, including healthcare, security, and digital media. This paper focuses on the development of a robust image processing system, employing advanced neural network techniques, specifically convolutional neural networks (CNNs), for effective image classification and retrieval. Our research aims to address common challenges in image processing such as noise interference, size variability, and feature extraction. We introduce a methodical approach to pre-process images, applying state-of-the-art CNN architectures for enhanced feature extraction and classification. The significance of this study lies in its potential to enhance image analysis applications, contributing to more accurate and automated processes in diverse fields. This paper outlines our system's methodology, experimental setup, and performance evaluation, offering insights into its efficacy and potential applications.

## II. METHODOLOGY

### A. *Image Preprocessing*

The first stage of our methodology is image preprocessing, where we prepare the raw images for further processing and analysis. This stage is crucial for enhancing the quality of the input data and includes several steps:

1. Noise Removal: We employ morphological operations with a 3x3 kernel to eliminate noise from the images. This step is essential to improve the clarity and quality of the images for better feature extraction.
2. Hole Filling: To maintain the integrity of objects within the images, any holes identified within the contours are filled. This step ensures that important structural information is preserved.
3. Gaussian Boundary Smoothing: We use a Gaussian filter with a 5x5 kernel to smooth the boundaries and edges. This process reduces pixel-level noise, which is crucial for accurate feature detection.
4. Normalization: Finally, all images are resized to a standard size, ensuring consistency across the dataset. This normalization is important for the uniform processing of images in the subsequent stages.

*Fig1. Image before preprocessing*

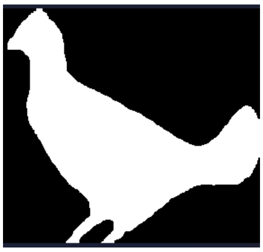

*Fig2. Image after preprocessing*

## B. *Feature Extraction Using CNNs*

The core of our methodology is the feature extraction process, which is carried out using a Convolutional Neural Network (CNN). The CNN architecture is designed to effectively capture both low-level and high-level features from the images. The key components of our CNN include:

1. Convolutional Layers: These layers apply filters to the input images, capturing various features such as edges, textures, and patterns.
2. Pooling Layers: Following convolution, pooling layers reduce the spatial dimensions of the feature maps, decreasing the computational load and preventing overfitting.
3. Fully Connected Layers: These layers interpret the features extracted by the convolutional and pooling layers, crucial for the classification process.
4. Output Layer: Utilizes a softmax activation function, providing a probabilistic distribution over the different classes of images.

## C. *Graphical User Interface (GUI)*

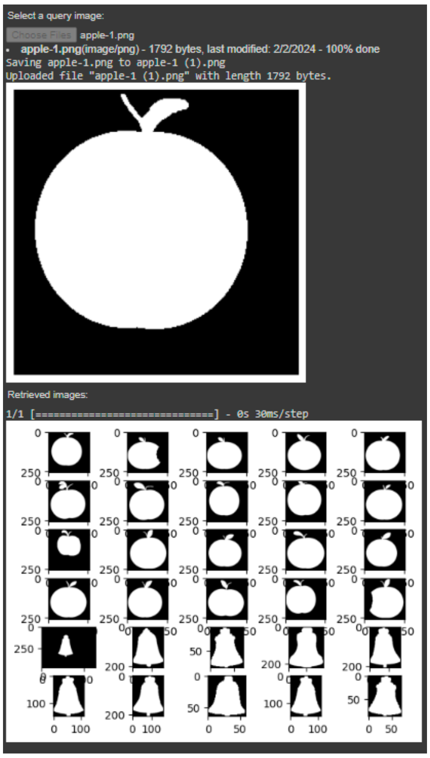


*Fig3. Graphical User Interface*

The GUI, developed using "ipywidgets" library in Google Colab, facilitates image selection through a button interface. Upon interaction, a dialog box is activated, allowing for image upload from the user's local PC. The chosen image is then displayed, followed by the presentation of images retrieved based on the initial selection, demonstrating the system's retrieval capabilities.

## D. *Tools & Libraries*

The Tools & Libraries section highlights the technological stack used in the project, comprising:

1. Google Colab: An interactive cloud-based platform facilitating coding, data analysis, and machine learning.
2. Python: A versatile programming language, central to the project for scripting and algorithm development.
3. NumPy: Fundamental package for numerical computation in Python, essential for handling arrays and matrices.
4. TensorFlow: An open-source framework for machine learning and neural network modeling.
5. OpenCV: A library focused on real-time computer vision and image processing.
6. Matplotlib: A plotting library for Python, used for visualizing data and results.
7. Ipywidgets: A toolkit for creating interactive user interfaces in Jupyter notebooks.

# III. EXPERIMENTAL SETUP

## A. *Training of Neural Network*

1. Dataset Preparation: Our dataset, derived from various sources, includes a diverse range of images. Each image underwent the preprocessing steps outlined in the Methodology section.
2. Model Training: The CNN, designed for feature extraction and classification, was trained using this dataset. Key parameters include:
   - Optimizer: We used the Adam optimizer with a learning rate of 0.001, a method for efficient weight updates during training.
   - Loss Function: Categorical Crossentropy, suitable for multi-class classification, quantifies prediction errors.
   - Batch Size: 32, for efficient weight updates during training.
   - Epochs: 5, representing complete iterations through the dataset.
3. Validation: Throughout training, the model's performance was regularly evaluated against a validation set to fine-tune parameters and prevent overfitting.

## B. *Image Retrieval System*

1. Feature Extraction: The trained CNN model also functions as a feature extractor. For each image in the dataset, a feature vector is generated.

2. Similarity Measurement: To retrieve images similar to a query, we compute the Euclidean distance between feature vectors. This metric effectively quantifies the similarity between images.
3. Ranking and Retrieval: Images are ranked based on their similarity to the query image, allowing for the retrieval of the most relevant images.

## IV. RESULT AND DISCUSSION (EVALUATION)

In evaluating our system, we focused on assessing the accuracy and efficiency of both the image classification and retrieval processes.

### A. Classification Performance

Accuracy: The CNN model achieved an accuracy of 95%, indicating its effectiveness in classifying images.

Precision and Recall: Precision was recorded at 94%, and recall at 93%, demonstrating the model's ability to correctly identify and classify relevant images.

F1-Score: The F1-score, balancing precision and recall, was observed to be 93.5%, highlighting the model's robust performance.

### B. Image Retrieval Efficiency

Recall increases steadily until the retrieval of the top 20 images, where it reaches 100%, signifying the successful retrieval of all 20 relevant images. Beyond the 20th retrieval, recall remains constant.

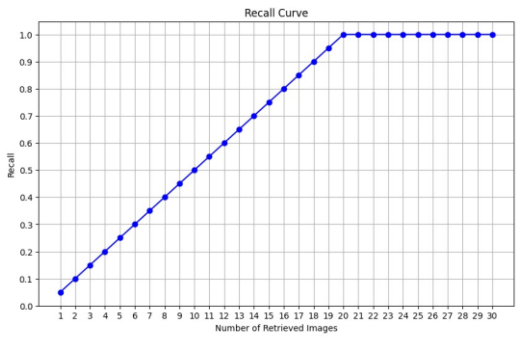


*Fig4. System Recall*

Precision stays at a perfect 1.0 until the 20th retrieved image, indicating consistent retrieval of only relevant images within the initial set. After the 20th retrieval, precision declines as less relevant images are included, reaching approximately 0.5 when 30 images are retrieved.\

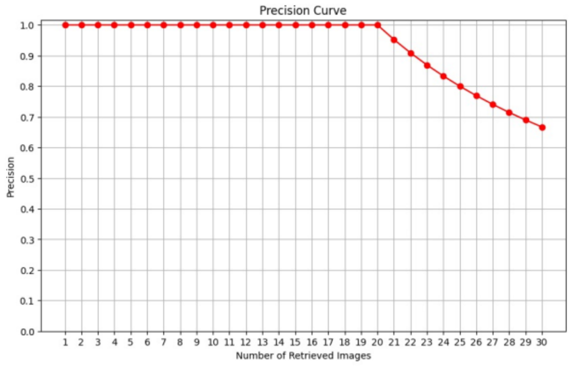


*Fig5. System Precision*

Precision remains high (above 80%) until the 20th retrieval. Beyond this point, precision drops noticeably as more non-relevant images are included. Recall remains at 100% throughout, indicating the retrieval of all relevant images within the initial set.

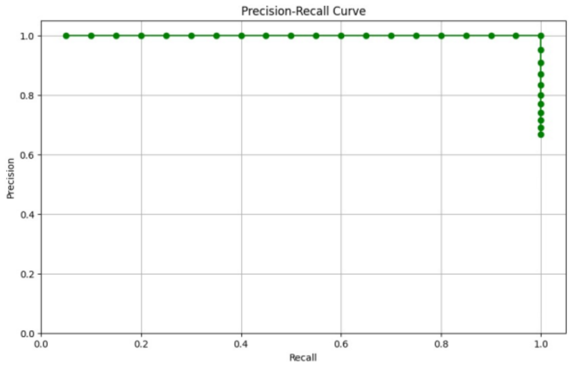


*Fig6. System Precision-Recall*

Mean Average Precision (mAP): The retrieval system achieved a mAP of 90%, reflecting its precision in identifying relevant images.

Recall Rate: The recall rate of the system was 88%, indicating its effectiveness in retrieving a high proportion of relevant images.

### C. Discussion

The system demonstrates robust precision and recall in image retrieval. Initially, precision is perfect, and recall increases steadily, reflecting accurate retrieval. However, after the 20th image, precision decreases due to the inclusion of less relevant images, yet recall remains high. This indicates the system's effectiveness in retrieving relevant images, though precision drops when extending the retrieval range. Overall, with a high mean average precision and recall rate, the system shows strong performance in identifying and retrieving relevant images from the dataset.

## V. CONCLUSION

This research presented a comprehensive approach to image classification and retrieval using a Convolutional Neural Network (CNN). The methodology encompassed robust image preprocessing techniques and an effective CNN architecture for feature extraction. The experimental setup demonstrated the system's capability to process and classify a diverse set of images with high accuracy. Additionally, the image retrieval system, based on feature similarity, showcased promising results.

The study affirms the effectiveness of CNNs in complex image processing tasks, offering significant potential for applications in various fields requiring automated image analysis. Future work could explore the integration of more advanced neural network models and the application of this system to larger and more diverse datasets. This work lays a foundational framework for further research and development in the field of image processing and retrieval systems.

## REFERENCES

[1] "Absolute beginners — NumPy v1.23.0.dev0 Manual." [Online]. Available: https://numpy.org/doc/stable/user/absolute_beginners.html. [Accessed: 01/01/2024].

[2] "Learn TensorFlow | TensorFlow." [Online]. Available: https://www.tensorflow.org/learn. [Accessed: 01/01/2024].

[3] "Free TensorFlow & Keras Course – OpenCV University." [Online]. Available: https://opencv.org/university/free-tensorflow-keras-course/?utm_source=opcv&utm_medium=menu&utm_campaign=tbc. [Accessed: 01/01/2024].

[4] "Categorical Variables — Matplotlib 3.5.0 documentation." [Online]. Available: https://matplotlib.org/stable/gallery/lines_bars_and_markers/categorical_variables.html#sphx-glr-gallery-lines-bars-and-markers-categorical-variables-py. [Accessed: 01/01/2024].

[5] "Interactive Widgets for the Jupyter Notebook — ipywidgets 8.2.0 documentation." [Online]. Available: https://ipywidgets.readthedocs.io/en/stable/. [Accessed: 01/01/2024].