WENZHOU-KEAN
UNIVERSITY

Multimodal Integration for Enhanced Image and text data classification and Retrieval

Bo Yu

Math 3790 W02

Dr. Ray-Ming, C

30 May, 2024

CONTENTS

1 Introduction 2

1.1 Motivation 2

1.2 Literature review 2

1.3 Scope 3

2 Problem and Models 4

2.1 Problem 4

2.2 Model methods 5

2.2.1 Grayscale image classification 6

2.2.2 Image classification with ResNet-50 7

2.2.3 Keyword extraction model 10

2.2.4 Combination 14

3 Experiment 14

3.1 Dataset 14

3.2 Pre-processing 16

3.3 Experimental Setup 17

3.4 Result 20

4 Evaluation and visualization 23

5 Conclusion 26

References 26

1 Introduction

1.1 Motivation

In today's era of information explosion, massive image and text data increase exponentially, how to effectively classify, correlate and retrieve these data has become an urgent problem to be solved. Traditional manual labeling and classification methods can no longer cope with such a huge amount of data, which not only increases the time and labor costs, but also limits the efficiency and accuracy of information retrieval. Especially in the scene described by landscape images and their related texts, users are eager to quickly find images closely associated with specific keywords, but the complex internal correlation between images and texts makes this demand difficult to achieve by simple means.

We need an intelligent solution, a system that can automatically analyze and process image and text data. It can not only classify images accurately, but also extract core information from text, and match and retrieve it accurately with images. Such a system can not only greatly improve the efficiency of data management and retrieval, but also provide users with more accurate and personalized service experience. Based on this background, this project aims to develop an intelligent image and text data processing system, which can not only automatically classify new landscape images, but also filter the matching landscape images according to the keywords extracted from the text content. To cope with the growing multimodal data processing needs of today's information society.

1.2 Literature review

Recent advances in deep learning and natural language processing (NLP) have significantly improved the capabilities of automated image and text classification systems. Convolutional neural networks (CNNs), particularly deep architectures like ResNet-50, have demonstrated remarkable performance in image classification tasks (He, Zhang, Ren, & Sun, 2016). The introduction of residual learning in ResNet-50 has addressed the

degradation problem in deep networks, enabling more effective training of very deep models by facilitating gradient flow through the network.

In the realm of text processing, techniques such as Term Frequency-Inverse Document Frequency (TF-IDF), Chi-Square Test, Pointwise Mutual Information (PMI), and Hidden Markov Models (HMM) have been widely used for keyword extraction and text classification (Rabiner, 1989). These methods allow for efficient representation and analysis of textual data by highlighting important terms and their associations. The integration of image recognition and NLP techniques has led to the development of multimodal systems capable of linking visual and textual data, thereby enhancing the retrieval and classification processes.

Several studies have explored the potential of combining image and text data for improved search and retrieval systems. For instance, Chaudhuri, Banerjee, Bhattacharya, and Datcu (2020) demonstrated the effectiveness of using deep learning models for cross-modal retrieval, where textual queries are used to retrieve relevant images. Similarly, Everingham, Van Gool, Williams, Winn, and Zisserman (2009) highlighted the benefits of using keyword extraction techniques in conjunction with image classification models to improve the accuracy and relevance of search results. These advancements underscore the feasibility and potential benefits of developing a system that leverages both deep learning for image classification and NLP for keyword extraction. Such a system can bridge the gap between textual and visual data, providing a more comprehensive approach to data management and retrieval in various applications.

1.3 Scope

This project aims to develop and evaluate a system that integrates deep learning-based image classification and NLP-based keyword extraction to enhance the classification, retrieval, and correlation of landscape images and their related text descriptions. The scope includes the following key components:

- Image Classification Models: Development and evaluation of two image classification models—one based on grayscale image processing with Principal Component Analysis (PCA) and another using the ResNet-50 deep convolutional

neural network.

- Keyword Extraction Model: Implementation of various keyword extraction techniques, including TF-IDF, Chi-Square Test, PMI, and HMM, to identify and extract significant keywords from text descriptions.
- Combination Model: Integration of the image classification and keyword extraction models to enable text-based image retrieval and correlation.
- Experimental Evaluation: Use of multiple datasets, including the Intel Image Classification Dataset, Flickr30k Dataset, and ImageNet Dataset Labels, to train, validate, and test the models. The performance of the models will be evaluated based on accuracy, precision, recall, and F1-score.

By addressing these components, the project aims to demonstrate the potential of combining deep learning and NLP techniques to create a robust and efficient system for multimodal data processing and retrieval.

2 Problem and Models

2.1 Problem

First, deep learning models have achieved remarkable results in image classification tasks, but classification for specific domains, such as landscape images, still faces challenges. Traditional image classification methods may not be able to accurately distinguish subtle differences, although gray level image recognition classification has the advantage of high computational efficiency, it may reduce the accuracy of classification due to the loss of important color information.

Secondly, in the image screening task based on text keywords, we are faced with the challenge of how to accurately extract scene-related keywords from the text and screen the corresponding pictures based on these keywords. This requires us to develop an effective NLP technique that can accurately understand the semantic information of the text and identify keywords related to the landscape.

Finally, combining image recognition and NLP technology, this requires us to be able to accurately extract not only the keywords in the text, but also to accurately identify the

features in the picture, and establish the association between the two. This is a more complex cross-modal matching problem, which requires us to develop an algorithm framework that can handle both text and image data, and overcome the semantic differences and presentation inconsistencies between the different modal data as much as possible.

2.2 Model methods

Variable	Description	Formula/Details
P_i	Flattened graph as a column vector	-
$\bar{P}$	Mean vector for all pictures	$\bar{P} = \frac{1}{M}\sum_{i=1}^{M} P_i$
M	Number of pictures	-
$\tilde{P}_i$	De-averaged vector after mean centralization	$\tilde{P}_i = P_i - \bar{P}$
C	Covariance matrix	$C = \frac{1}{M}\sum_{i=1}^{M} \tilde{P}_i \tilde{P}_i^T$
λ_k	Eigenvalue of the covariance matrix	-
v_k	Eigenvector corresponding to the eigenvalue λ_k	-
W	Projection matrix formed from the eigenvectors corresponding to the largest eigenvalues	-
Y	Projected data in the principal component space	$Y = W^T \tilde{P}$
Z	Projection coordinate for new samples	-
d	Euclidean distance between new sample and training set	-
Y_i	Nearest principal component space	-
$H(x)$	Target map	-
$f(x)$	Output	-
$F(x)$	Residual function	$F(x) = H(x) - x$
y	Final output	$y = F(x) + x$
x	Input	-
$TF_{i,j}$	Term Frequency of word w_i in document d_j	-
N	Total number of documents	-
χ^2	Chi-square value for keyword extraction	$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$

O_i	Observed frequency	-
E_i	Expected frequency	-
$PMI(w_1, w_2)$	Pointwise Mutual Information between words w_1 and w_2	$PMI(w_1, w_2) = \log \frac{P(w_1,w_2)}{P(w_1)P(w_2)}$
$P(w_1, w_2)$	Probability that words w_1 and w_2 occur simultaneously	-
$P(w_1)$	Probability that word w_1 occurs independently	-
$P(w_2)$	Probability that word w_2 occurs independently	-
S	State set for HMM	$S = \{s_1, s_2, \ldots, s_n\}$
O	Observation set for HMM	$O = \{o_1, o_2, \ldots, o_m\}$
π	Initial probability distribution for HMM	-
A	State transition probability matrix for HMM	-
B	Observation probability matrix for HMM	-
s_t	System state at time t	-
o_t	Observation at time t	-

2.2.1 Grayscale image classification

Grayscale image classification based on principal component analysis (PCA) can reduce the data dimension by dimensionality reduction. Using as few dimensions as possible, data information can be kept as much as possible, noise and redundant information can be removed, and computation efficiency and classification performance can be improved.

- Assuming that the original graph is an $n * l$ matrix, flatten each graph so that it becomes an $n * l$ column vector P_i.
- Compute the mean vector $\overline{P}$ for all pictures P_i, where M is the number of pictures:

$$\overline{P} = \frac{1}{M}\sum_{i=1}^{M} P_i$$

- Take the mean centralization, subtract the mean vector P from each image, and get the de-averaged vector $\widetilde{P_i}$:

$$\tilde{P}_i = P_i - \overline{P}$$

- After centralization, the covariance matrix C is calculated through the outer product, which is to reflect the linear relationship between the various features in the data:

$$C = \frac{1}{M}\sum_{i=1}^{M} \tilde{P}_i \tilde{P}_i^T$$

- By solving the eigenvalue λ_i and the eigenvector v_i of the covariance matrix, the

eigenvector corresponding to the first k largest eigenvalues is selected to form the projection matrix:

$$Cv_i = \lambda_i v_i$$

- When the centralized data is projected into the principal component space W, the projected data Z retains the most important features in the low-dimensional space:

$$Z = W^T \tilde{P}$$

- Then, in the face of the new sample N, the matching can be completed and the projection coordinate N_e can be calculated:

$$N_e = W^T(N_e - \bar{P})$$

- Match the Euclidean distance $\hat{k}$ between the new sample N_e and the training set $P_{j,B}$ to complete the classification requirements:

$$\hat{k} = \arg\min_j \|N_e - P_{j,B}\|^2$$

- According to the number of categories i, there are i different principal component Spaces. Select the smallest d, that is, the nearest principal component space W_i, to complete the classification of samples.

2.2.2 Image classification with ResNet-50

ResNet mainly solves the "degradation" problem of deep convolutional networks when the depth deepens. In general convolutional neural networks, the first problem caused by increasing the depth of the network is gradient disappearance or gradient explosion. The core idea is to introduce residual connections so that the network can be trained more deeply and alleviate the problem of disappearing gradients. Specifically, assume that the target map is $H(x)$, output is x, residual blocks learn:

$$F(x) = H(x) - x$$

And final output:

$$F(x) + x$$

ResNet-50 is one of these deep convolutional neural networks. The ResNet-50 contains 50 layers of depth and lets the network learn the Residual between the input and

the target output through a residual connection, rather than learning the target mapping directly. This design makes training deep networks much easier and more stable.

In ResNet-50, the network is stacked with multiple Residual blocks, each containing several convolutional layers and a Skip Connection between input and output. This residual block structure allows the network layer to pass information through this shortcut, which effectively alleviates the problem of information loss and gradient disappearance in deep networks (Fig. 1). In particular, the ResNet-50 introduces a bottleneck design that reduces and recovers the number of channels through a 1×1 convolution layer, thereby reducing computational effort while maintaining efficient expression capabilities.

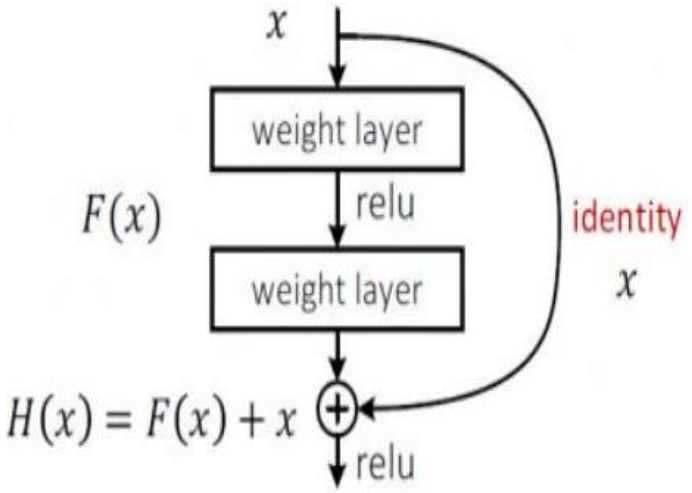


Figure 1. Residual learning: a building block.

In particular, the ResNet-50 introduces a bottleneck design that primarily uses three-layer residual cells for deeper networks. A three-layer residual unit, also known as a bottleneck structure, firstly reduces dimension with a 1*1 convolution, and finally restores the original dimension with 1*1 elevation (Fig. 2). Thus, the computational workload is reduced while maintaining efficient expression ability. If the input and output dimensions are different, you can do a linear mapping of the input to transform the dimensions, and then connect the following layers. The three-layer residuals reduce the number of parameters for the same number of layers, so that deeper models can be expanded.

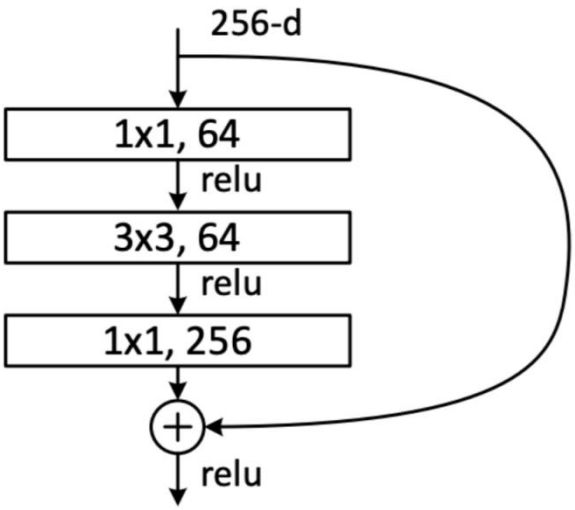


Figure 2. Bottleneck structure

In the training process, the image is preprocessed first, and the generalization ability of the model can be improved by data enhancement (such as random horizontal flipping and random rotation). Image standardization is also an important step, which needs to be normalized according to the mean and standard deviation of the pre-trained model. Model training uses the cross entropy loss function to measure the difference between the model output and the real label, adopts the AdamW optimizer, improves the training efficiency by adaptive learning rate adjustment, and introduces weight attenuation to prevent overfitting. The learning rate scheduler gradually reduces the learning rate at the end of the training period to fine-tune the model parameters.

When trained, ResNet-50 consists of four main stages (Stage1 to Stage4), each consisting of multiple residual blocks, and the first residual block of each stage may undergo downsampling operations (Fig. 3). To be specific:

- Stage1: Contains 3 residual blocks, the first residual block is downsampled using a 3x3 convolution layer with step size 2, and the size of the output feature map is halved.
- Stage2: Contains four residual blocks. The first residual block is also downsampled.
- Stage3: Contains 6 residual blocks, the first residual block is downsampled.
- Stage4: Contains 3 residual blocks. The first residual block is downsampled.

- Output layer: After Stage4, a Global Average Pooling layer is usually added to turn the feature map into a feature vector. The feature vectors are then mapped to specific classification results through a Fully Connected Layer.

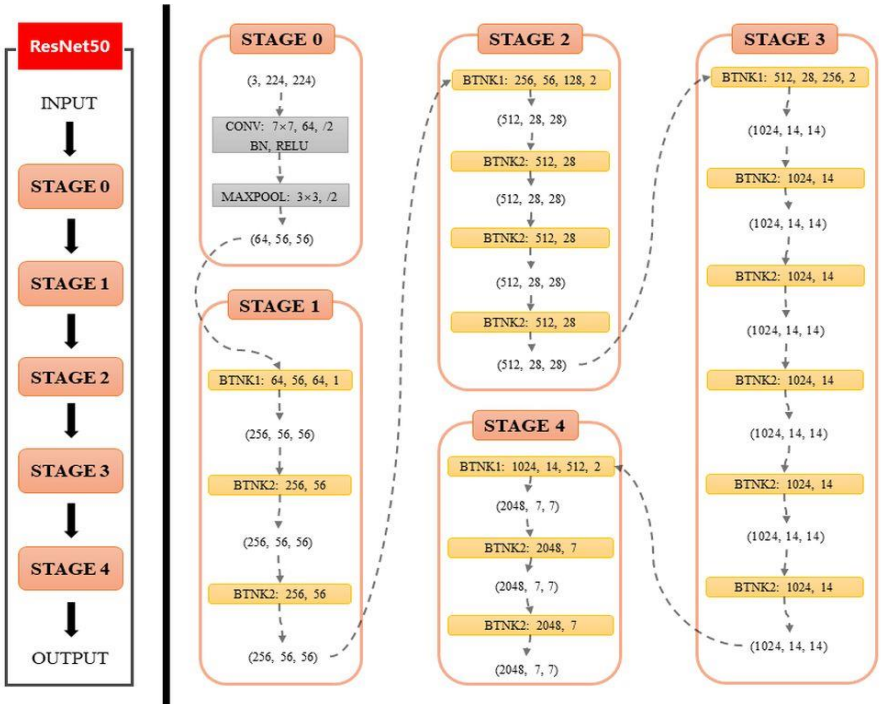


Figure 3. Network structure of ResNet-50

In order to adapt to different tasks, the output layer of the ResNet-50 can be adjusted according to specific needs. For image classification tasks, the number of output nodes can be set to the number of categories. With this flexible adjustment, combined with effective data enhancement strategies and hyperparameter optimization, the ResNet-50 performs well in a variety of image recognition and classification tasks with strong performance and reliability.

2.2.3 Keyword extraction model

Hoping to extract keywords from a set of similar text descriptions, and you can do this in batches in text sets.

1. TF-IDF (Word frequency - inverse document frequency)

TF-IDF is a common text representation used to measure the importance of words in a document. While TF represents the frequency with which a word appears in a document, IDF measures the word's general importance, i.e. its rarity in the overall document collection.

Formula:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t)$$

Where $\mathrm{TF}(t,d)$ is a word t is in the documentation, the frequency of occurrence in d, $\mathrm{IDF}(t)$ represents the inverse document frequency of the word t, defined as:

$$IDF(t) = \log\left(\frac{N}{DF(t)}\right)$$

Where N is the total number of documents and $\mathrm{DF}(t)$ is the number of documents containing the word t.

2. Chi-Square Test

A Chi-square test is a statistical method used to detect a correlation between two categorical variables. In keyword extraction, the Chi-square test is used to measure the strength of the association between each word and the category label.

Formula:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where O_i is the observed frequency and E_i is the expected frequency. By calculating the chi-square value of each word, it is possible to determine the relevance of the word to the target category.

3. PMI (Point Mutual Information)

PMI measures the correlation between the probability of two words appearing together and the probability of them appearing independently. A high PMI indicates a strong correlation between the two words.

Formula:

$$\mathrm{PMI}(x,y) = \log \frac{\mathrm{P}(x,y)}{\mathrm{P}(x)\mathrm{P}(y)}$$

Where $P(x,y)$ is the probability that words x and y occur simultaneously, and $P(x)$ and $P(y)$ are the probability that words x and y occur independently, respectively.

4. Hidden Markov model (HMM)

A Hidden Markov model (HMM) is a statistical model used to model time series data or serial data. HMM assumes that the system is a Markov process (the current state depends

only on the previous state) and that the state is not visible (implied) but can be inferred from observations, Below is a Markov time series (Fig. 4).

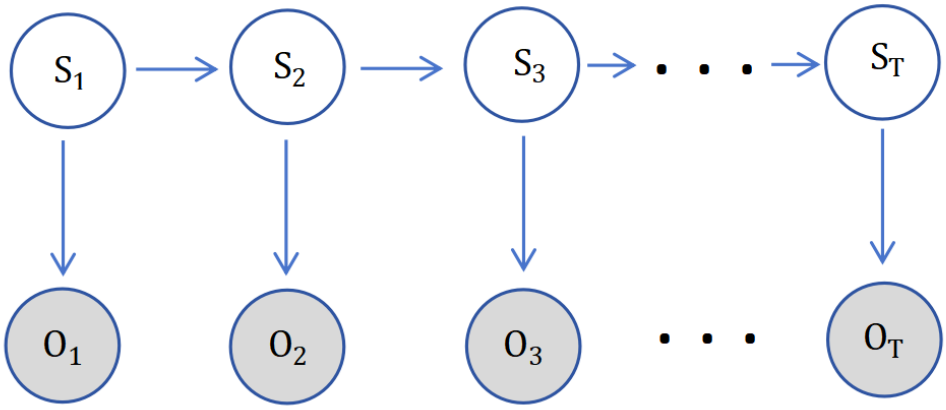


Fig. 4 Markov time series

The definition of HMM is based on two assumptions:

Hypothesis 1: Suppose that the state of the hidden Markov chain at any time t depends only on the state at the previous time $(t-1)$, independent of the state and observations at other times, and independent of time t.

Formula:

$$p(s_t \mid s_{t-1}, o_{t-1}, \ldots, s_1, o_1) = p(s_t \mid s_{t-1})$$

Where S_t is the system state at time t.

Hypothesis 2: The observation at any time is assumed to depend only on the state of the Markov chain at that time, independent of other observations and states.

Formula :

$$p(o_t|s_r, \ldots, s_{t+1}, o_{t+1}, s_t, o_t, s_{t-1}, o_{t-1}, \ldots, s_1, o_1) = p(o_t|s_t)$$

Where O_t belongs to O and represents the observed value of t.

In the keyword extraction task, HMM can be used to capture the sequence information of words, so as to identify keywords. Its basic components include:

- State set (S) : defined as "keyword" (K) and "non-keyword" (N).
- Observation set (O) : All words in the text.
- Initial probability distribution (π) : Sets the probability that the system is initially in a "keyword" or "non-keyword" state. For example, you can assume that 20% of the words in the text are keywords.

- State transition probability matrix (A) : Defines the probability of moving from one state to another. In general, you can assume transition probabilities between keywords, between non-keywords, and between keywords and non-keywords. For example:

$$A = \begin{bmatrix} P(K \to K) & P(K \to N) \\ P(N \to K) & P(N \to N) \end{bmatrix}$$

- Observation probability matrix (B) : Defines the probability of generating a word in a certain state. For example:

$$B = \begin{bmatrix} P(\text{word}_1 \mid K) & P(\text{word}_2 \mid K) & \dots \\ P(\text{word}_1 \mid N) & P(\text{word}_2 \mid N) & \dots \end{bmatrix}$$

These methods combine to extract keywords from text. First, the input text set needs to be preprocessed, including the removal of stops and word segmentation. Next, the importance of each word is calculated using the TF-IDF method, and the relevance of each word to the "target" category is evaluated using the Chi-square test, so as to select a certain number of keywords that rank in the top. The correlation between the word pairs is then mined through the PMI calculation, and the initially extracted keywords are mapped to the predefined category mapping set (target category: relevant words can be defined as terms of that category). Finally, in order to analyze these texts more deeply, a hidden Markov model (HMM) is constructed. HMM can capture the hidden state sequence in the text and generate a new state sequence or predict the potential structure of the text by learning the current text sequence.

By combining TF-IDF, Chi-square test and PMI, the key information in the text can be analyzed and mined from different angles. The application of these methods in keyword extraction and classification improves the accuracy and relevance of keyword recognition. The introduction of HMM model provides a powerful tool for further exploring the hidden structure of text and generating new text. Through this combination method, we can understand the text content more comprehensively and deeply, and lay a foundation for the subsequent research on the combination of image text.

2.2.4 Combination

This model shows how to combine Image recognition and natural language processing technology, use Keyword extraction model to extract text keywords and match them with the feature tags after Image recognition model of ResNet-50. Thus, it can screen out related images according to text keywords.

The pre-trained ResNet-50 model is used to extract and classify the features of the images. Through image preprocessing, the image is adjusted to the appropriate size and normalized, and then the image is input to the ResNet-50 model to obtain the corresponding image label. These tags are based on the ImageNet dataset and cover a large number of common objects and scenes. Next, the text data is preprocessed, including stop word removal and word segmentation, and then the importance of each word in the text is calculated by TF-IDF method to extract keywords. After the keywords are obtained, the Hidden Markov model (HMM) is used to further analyze the text sequence, and the most representative keywords are identified and extracted. Finally, by matching these keywords and image tags, we can find the images associated with these keywords and complete the text-based image screening and matching. It can also be applied to image search, recommendation, classification, annotation, cross-modal retrieval and other applications to make the association between text and image more accurate and efficient.

3 Experiment

3.1 Dataset

Three different datasets were employed in this study: the Intel Image Classification dataset, the Flickr30k dataset, and the ImageNet dataset label, each of which was selected for its unique contribution to various aspects of the image analysis and classification task. And a keyword mapping table.

Intel Image Classification Dataset

The Intel Image Classification Dataset (Bansal, 2019) is specifically designed to evaluate image classification models. This data set contains approximately 25k images of size

150x150, distributed in six categories: buildings, forests, glaciers, mountains, oceans, and streets. Each class contains a different set of images captured under different conditions, providing a solid foundation for training and validating image classification algorithms. The dataset is divided into three subsets: training, validation, and testing to ensure comprehensive evaluation and model development. The diversity of scenario types and environmental conditions represented by this dataset enhances the generalization of classification models developed using it.

The Flickr30k Dataset

The Flickr30k Dataset (Shawn, 2023) is an extensive collection of images from Flickr for image captioning and visual description tasks. It contains 31,783 images that capture people engaged in everyday activities and events. Each image has five descriptive titles provided by human annotators, making the dataset ideal for training models that require a nuanced understanding of visual content and natural language. Annotations in the Flickr30k dataset support supervised learning tasks designed to improve the accuracy of the image captioning system. In addition, the diversity of images, covering a wide range of everyday scenes and activities, helps to develop the robustness of the model.

ImageNet Dataset Labels

The ImageNet Dataset Labels (Athalye, 2019) is derived from the ImageNet Large-scale Visual Recognition Challenge (ILSVRC), a benchmark in the field of computer vision. With more than 14 million images grouped into 1,000 categories, ImageNet offers unparalleled scale for training and evaluating image recognition models. Each image in the dataset is labeled with a precise object category, facilitating the development of high-performance models that can distinguish between various objects. The tags provided by ImageNet help with supervised learning, enabling detailed and accurate model training. The wide variety and volume of labeled data available in the ImageNet dataset is critical to achieving state-of-the-art performance in image classification tasks.

Landscape_keyword Mapping Table

A mapping table for landscape keywords is in Table 1

Landscape Category	Keywords
Building	building, house, cabin, church

Landscape Category	Keywords
Forest	forest, woods, trees, jungle
Glacier	glacier, ice, snow, arctic
Mountain	mountain, hill, peak, summit
Sea	sea, ocean, beach, shore
Street	street, road, path, alley

Table 1: Landscape_keyword Mapping Table

3.2 Pre-processing

The data processing flow for this study includes several careful steps to ensure that the data set is adequately prepared for the training and evaluation of image classification and keyword models. Each data set is preprocessed specifically based on its characteristics and the requirements of the task at hand.

For Intel image classification datasets, preprocessing begins by adjusting all images to a uniform size, typically 224×224 pixels, to ensure consistency in the dataset. This step is critical because it standardizes the input size of the neural network model, which improves training efficiency and model performance. After resizing, the image is normalized by scaling the pixel values to a range between 0 and 1. Normalization helps to accelerate the convergence of training process and enhance the stability of the model. Data enhancement techniques such as random rotation, flipping, and scaling are also applied to manually expand data sets and improve the robustness of the model to the various transformations and distortions that occur in real-world scenarios.

For the Flickr30k dataset, pre-processing steps are required to accommodate visual and text data. Similar to the Intel dataset, the images are resized and normalized. In addition, text titles are preprocessed by labeling sentences, converting them to lower case, and removing punctuation. These steps are essential for creating clean and standardized inputs for natural language processing models. In addition, word embedding and single thermal coding techniques are used to encode the title into a numeric format, enabling it to be fed into the neural network. This dual preprocessing ensures format compatibility of

visual and text data and promotes effective training of image captioning models.

For ImageNet dataset labels, pre-processing involves mapping the original image labels to a simplified set of categories to reduce complexity. This step utilizes a set of predefined keywords associated with each category. For example, labels such as "building," "house," "cottage," and "church" are grouped under the broader category of "building." This classification is based on predefined landscape keyword mappings. Simplifying labels helps reduce the complexity of the classification task and helps the model focus on distinguishing between a manageable number of high-level categories. In addition, labels are encoded into numerical values, which is a prerequisite.

A pre-processing pipeline for all datasets ensures that the data fed to the model is clean, consistent, and properly formatted for a specific task.

3.3 Experimental Setup

In experiments with grayscale image classification, we want to resize the image to 64×64 pixels and convert it to grayscale image. After calculating the covariance matrix, the eigenvalues and eigenvectors are obtained by eigenvalue decomposition method (np.linalg.eigh). After repeated debugging, the eigenvector corresponding to the first 39 maximum eigenvalues is selected as the principal component and projected into the principal component space (Fig. 5) for dimensionality reduction. Then, the Euclidean distance is used for classification. Select the label of the nearest training image as the prediction result. In order to evaluate the model, we calculate the classification accuracy through confusion matrix and classification report, and analyze the classification performance in detail. The confusion matrix (Fig. 6) is generated using the confusion_matrix and ConfusionMatrixDisplay functions, and the classification report is generated using the classification_report function. These metrics help us understand how the model performs in different categories.

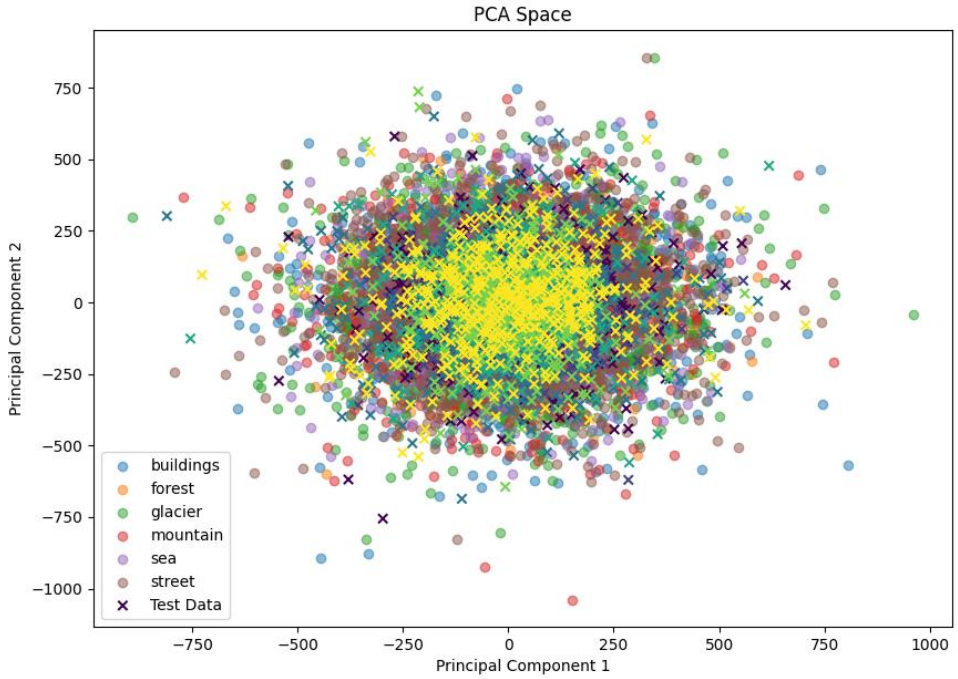


Figure 5. Principal component space

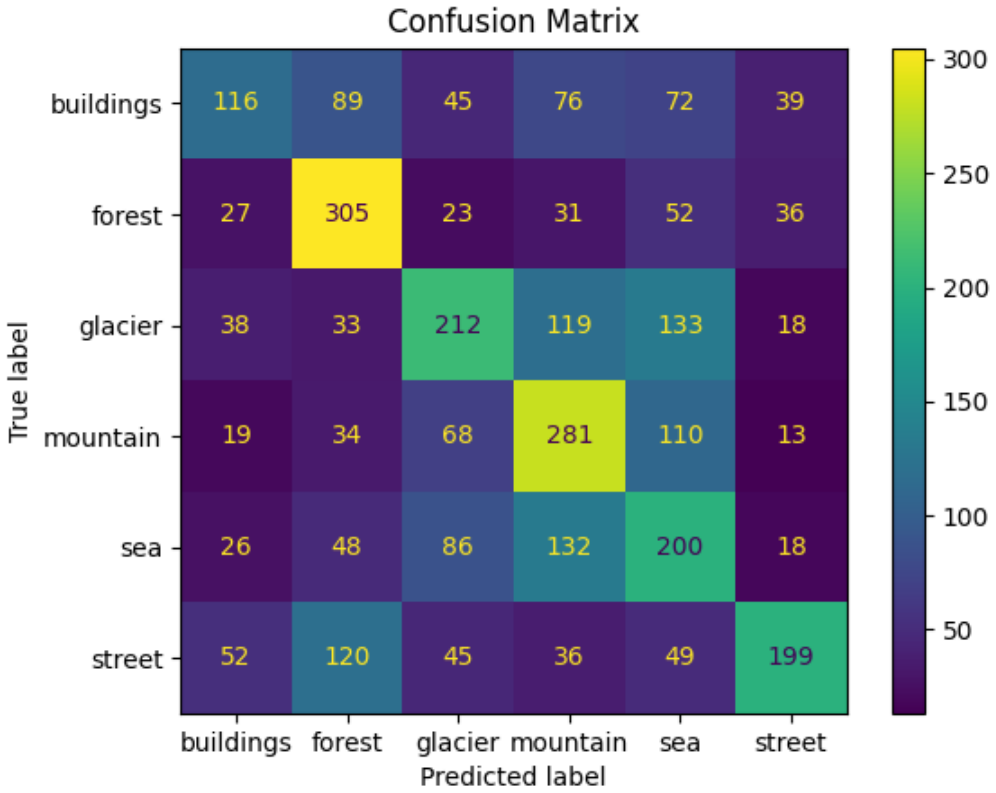


Figure 6. Confusion matrix

In experiments with ResNet-50's image classification model, we first performed random horizontal flipping and random rotation on the training data to improve the

generalization ability of the model, and all images were adjusted to 224×224 pixels and normalized processing. A pre-trained ResNet-50 model was used to match the number of output categories in our dataset by adjusting the last fully connected layer. We used the AdamW optimizer, which set the learning rate to 0.001, and the learning rate scheduler, which attenuates the learning rate by a factor of 0.1 every 5 cycles.

The model was trained over 20 cycles, and in each cycle, we propagated forward, calculated losses, backpropagated, and updated the parameters on the training set. Training losses and test accuracy were recorded at the end of each cycle. To evaluate the model in depth, we plotted the changes in loss and accuracy during training, visualized the confusion matrix to see how the model performed in each category, and generated a classification report detailing accuracy, recall, and F1 scores in each category. In addition, we also use t-SNE algorithm to reduce the dimension of features and visualize the distribution of features in two-dimensional space in order to more intuitively observe the distribution of different categories in the feature space.

In the experiment of the keyword extraction model, we first preprocess the text description of the Flickr30k dataset, remove the stop words and standardize them to lower case. Then TF-IDF vectorization method and Chi-square test were used to extract and screen the top 500 keywords, and these keywords were used to identify the landscape category keywords. Keywords map to specific categories, such as "building", "forest", "glacier", and so on. We then used the HMM model to model the text sequence. By converting the text description into an ASCII sequence, we trained the HMM model to generate similar text sequences to help us judge keywords. Finally, according to the position of these keywords in the description text, we extract the corresponding image ID and description. In order to organize and display the results, we show the distribution of various keywords through the histogram, and also show the keywords selected by TF-IDF and Chi-square test through the word cloud.

In this combined experiment, we used a pre-trained ResNet-50 model to make label predictions on images. The TF-IDF method is used to measure the importance of words. By labeling all words in the text and training a polynomial hidden Markov model, we calculate the generation probability of each word and select the word with the highest

generation probability as the keyword. Then matching these keywords with image tags, we find the images associated with these keywords and display them. Finally, when evaluating the performance of the model, we calculated and visualized the confusion matrix and generated a categorised report with metrics such as accuracy, recall, and F1 scores to fully evaluate the model's performance. Through these steps, we can extract meaningful keywords from text descriptions and find pictures related to these keywords in a large number of images, thus verifying the effectiveness of this method in image and text matching.

3.4 Result

The grayscale image classification approach using PCA achieves a modest classification accuracy of 43.77%, with noticeable fluctuations in precision, recall, and F1-score across different classes and epochs. The visualizations indicate instability in the training process, suggesting that the model struggles to effectively capture and generalize the features from the grayscale images, resulting in inconsistent performance. The results of the classification statistics are as follows: Figure 7 is the random display results of the five categories, and Figure 8 is the average Euclidean distance for each class.

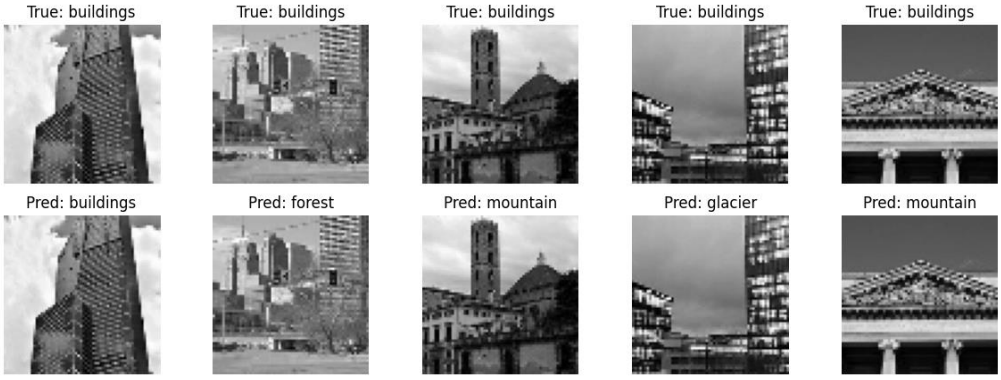


Figure 7. Prediction result presentation

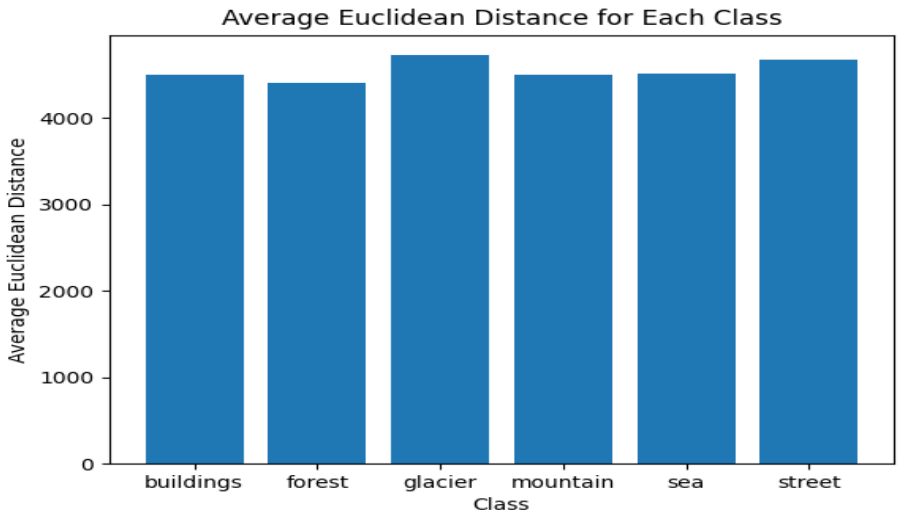


Figure 8. Average Euclidean distance histogram for each class

In contrast, the ResNet-50 model demonstrates significantly better performance, with accuracy, precision, and recall metrics stabilizing above 92% after the initial training epochs. The visualizations show a clear downward trend in loss and consistently high accuracy, precision, and recall, highlighting the model's robust learning and generalization capabilities. This stark difference underscores the superiority of the deep learning approach of ResNet-50 over the PCA-based method for image classification tasks.

About the model of landscape keyword extraction, the image ids and descriptions containing landscape keywords are screened by keyword mapping table and keyword extraction. Then, through mapping, the selected images about the landscape are found from the image set. We generate a category histogram (Fig. 6) from which we can visually see the distribution of keywords in each category, and also show the word cloud (Fig. 7) for the top 50 keywords of the Chi-square test.

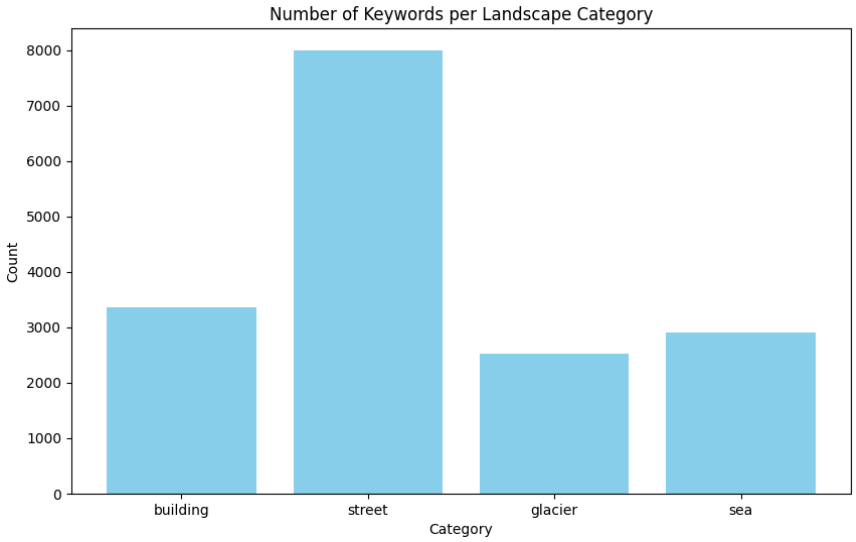


Fig. 6 Category keyword distribution histogram

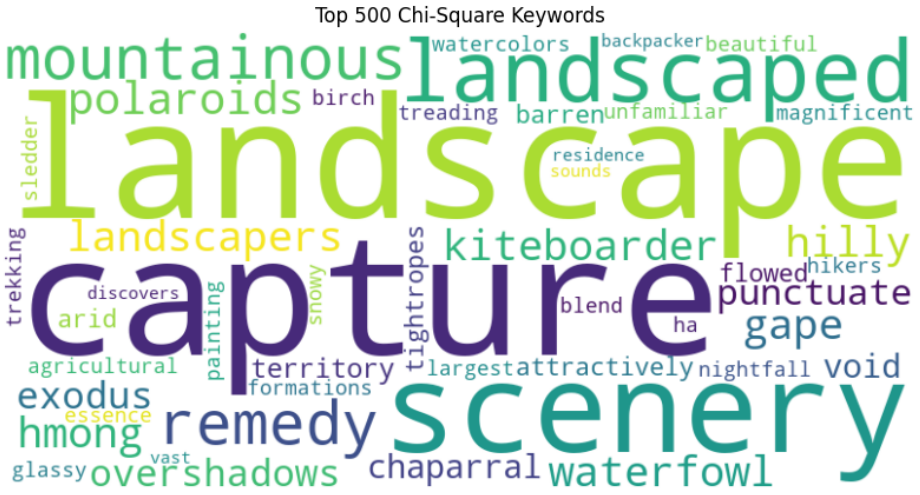


Fig. 7 The top 50 word cloud of the Chi-square test

Using ResNet-50 for image recognition allows for the extraction of detailed image features, which can then be matched with textual descriptions using TF-IDF and HMM models to find relevant images. For the given text descriptions (Fig. 8) such as "The bird flies in the forest," and "A serene lake surrounded by forest," keywords like 'forest', 'flies', and 'bird' were identified and used to retrieve images. This advanced approach effectively bridges the gap between textual content and visual data, making it possible to find images that match specific textual descriptions based on the extracted features.

```
"The bird flies in the forest.",
"A serene lake surrounded by forest.",
"The beach is full of tourists.",
"The desert is vast and dry.",
"A river flows through the valley.",
"The sky is clear and blue."
```

Fig. 8 Text descriptions

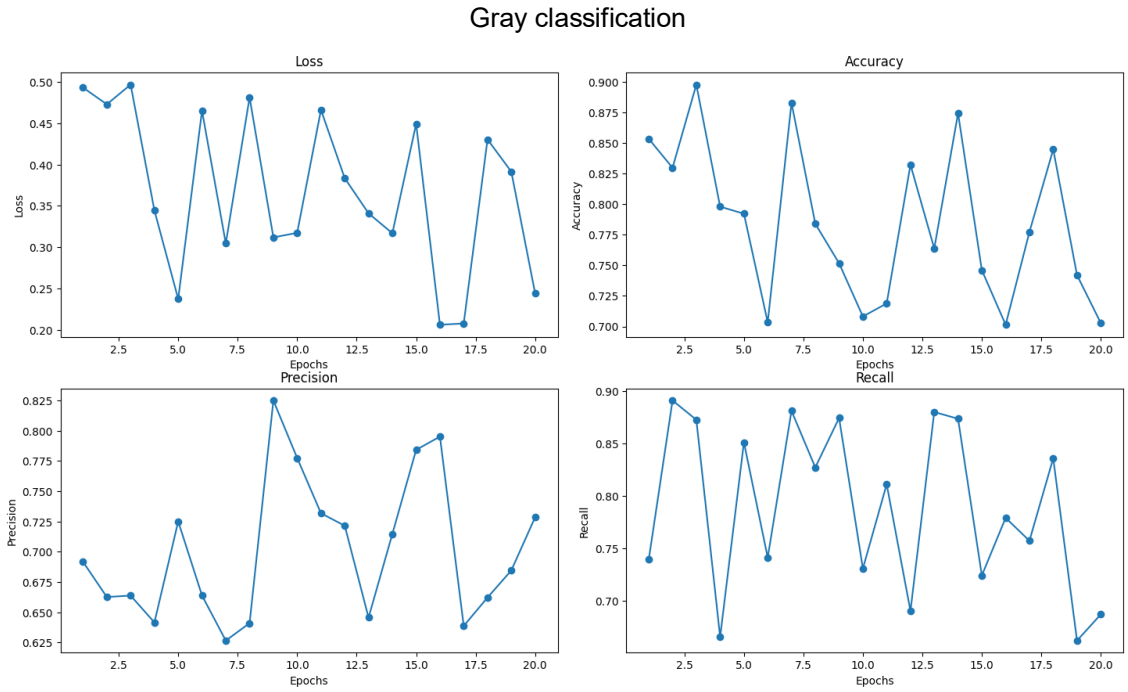
The method successfully identified and matched nine images (Fig. 9) that correspond to the descriptive keywords, demonstrating its ability to accurately associate text with visual content. This showcases the power of combining deep learning techniques with traditional text analysis models to achieve a sophisticated level of multimedia information retrieval. The results highlight the potential for such integrated approaches to enhance the efficiency and accuracy of search and retrieval systems in various applications.

Fig. 9 Matching images

4 Evaluation and visualization

The comparison between the grayscale image classification model and the ResNet-50 classification model reveals significant differences in their performance and complexity. The grayscale model, which involves reducing the images to grayscale, resizing, flattening, and using PCA for feature extraction, achieved a classification accuracy of 43.77%. The model's evaluation metrics show a precision of 0.45, recall of 0.44, and

F1-score of 0.43 across various classes, with particularly low performance in classifying 'buildings' and 'sea' images. The visualizations indicate high variability in loss and accuracy across epochs, reflecting instability and limited learning capability from the grayscale features. On the other hand, the ResNet-50 model, a deep convolutional neural network, significantly outperformed the grayscale model. Training loss decreased consistently over epochs, and the model achieved high accuracy, precision, and recall values from early epochs, stabilizing around 92% accuracy. This model's advanced architecture allows for more effective feature extraction and learning from complex image data. The evaluation results show a robust and reliable performance, with higher consistency in the precision and recall metrics across epochs. Figure 10 below is a visual comparison of the above two evaluations, including loss, accuracy, precision and recall. Visual comparisons of the training process indicate that ResNet-50 is far superior in handling and learning from image data, offering a more sophisticated and effective approach to image classification tasks.



ResNet-50

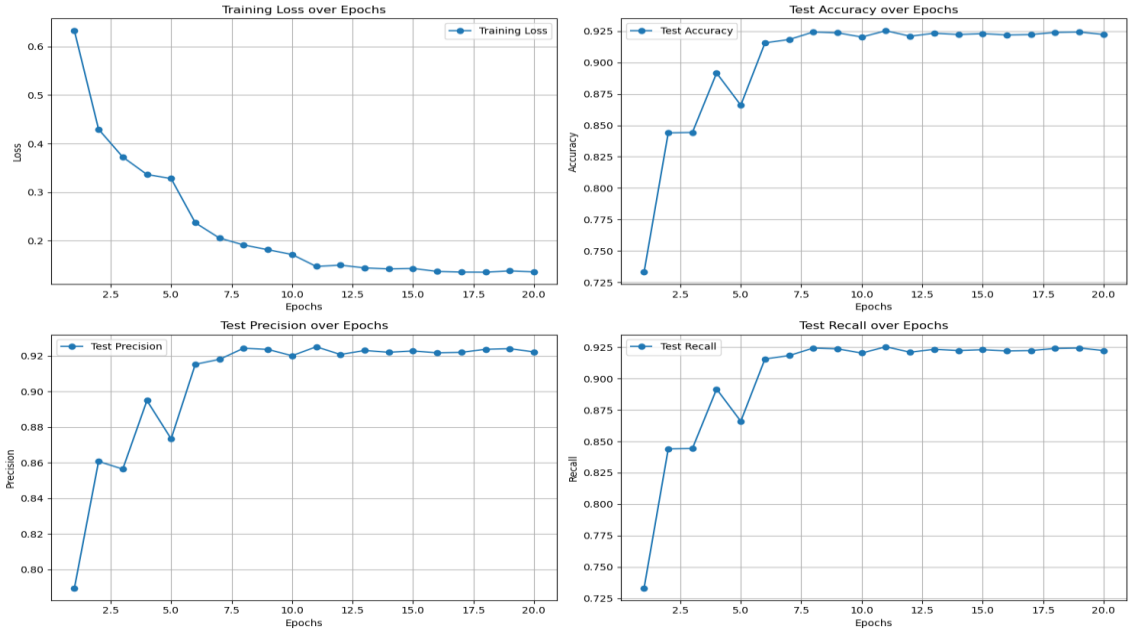


Fig. 10 Comparison

The landscape keyword extraction model successfully retrieves images matching these keywords, which proves its effectiveness. This can be seen in the precise retrieval of images associated with the identified keywords, which provides a practical method for linking text and visual data. However, the model may have difficulty understanding the context of the keywords in the description. For example, it may misunderstand a complex scene described by multiple elements together, resulting in less accurate image retrieval. At the same time, as the data set grows, so does the complexity of maintaining and updating the keyword mapping table. The model may require significant computational resources and optimization to efficiently handle larger, more diverse data sets.

Finally, the model approach, which combines image recognition and keyword selection, demonstrates the power of combining deep learning with traditional text analysis, providing a high level of accuracy in associating text content with visual data. The ability to bridge between text and images highlights the potential of this technology to enhance search and retrieval systems in a variety of applications. However, this comprehensive approach also presents some challenges. One of the main limitations is the reliance on the quality and specificity of textual descriptions; Vague or ambiguous descriptions can lead to reduced accuracy in image retrieval. In addition, the computational complexity and resource requirements for training deep learning models

such as ResNet-50 can be very large, which can limit their accessibility in small-scale applications. Despite these shortcomings, the combination of deep learning and text analysis models offers a promising direction for improving the efficiency and accuracy of multimedia information retrieval systems.

5 Conclusion

By comparing the gray-scale image classification model and ResNet-50 model, we can see the significant differences in performance and complexity between them. Gray-scale models using PCA for feature extraction have low accuracy and show high variability in terms of loss and accuracy, indicating the instability and limited learning ability of gray-scale features. In contrast, the deep convolutional neural network ResNet-50 model showed higher accuracy, precision, and recall rates, stabilizing at around 92% early in training. The model's advanced architecture enables efficient extraction of features from complex image data, enabling robust and reliable performance across a variety of categories.

In addition, the landscape keyword extraction model can effectively retrieve images matching the recognized keywords, which shows its practicability in linking text and visual data. However, as the dataset grew, the model faced challenges in understanding complex scenarios described by multiple elements and maintaining keyword mappings.

Models that combine image recognition and keyword selection highlight the potential for combining deep learning with traditional text analysis to enhance search and retrieval systems. Despite the challenges of relying on high-quality text descriptions and large computational requirements, this combined approach offers a promising direction for improving the efficiency and accuracy of multimedia information retrieval.

References

Athalye, A. (2019). imagenet-simple-labels. Retrieved 2024, from Githubusercontent.com website: https://raw.githubusercontent.com/anishathalye/imagenet-simple-labels/master/imagenet-simple-labels.json

Bansal, P. (2019). Intel Image Classification. Retrieved 2024, from www.kaggle.com website: https://www.kaggle.com/datasets/puneet6060/intel-image-classification

Chaudhuri, U., Banerjee, B., Bhattacharya, A., & Datcu, M. (2020). CMIR-NET : A deep learning based model for cross-modal retrieval in remote sensing. *Pattern Recognition Letters*, *131*, 456–462. https://doi.org/10.1016/j.patrec.2020.02.006

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2009). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, *88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/cvpr.2016.90

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286. https://doi.org/10.1109/5.18626

Shawn. (2023). Flickr30k. Retrieved June 2, 2024, from www.kaggle.com website: https://www.kaggle.com/datasets/eeshawn/flickr30k?select=flickr30k_images

Code:https://github.com/ybohhh/3790-Project-Multimodal-Integration-for-Enhanced-Image-and-text-data-classification-and-Retrieval.git