A Report on Mainstream Image Recognition Technologies in Computer Vision

Liu Ziqi
2021300130
Northwestern Polytechnical University
1114709266@qq.com

Abstract

The purpose of this paper is to present a report and review of fundamental concepts in the field of computer vision, including the application of traditional image descriptors (such as HoG, SIFT, etc.) in tasks such as image recognition, as well as the application of deep learning methods (CNN, RNN, Transformer, etc.) and transfer learning in computer vision tasks such as image recognition.

1. Introduction

The field of computer vision has experienced explosive development in recent years, transitioning from manual feature extraction methods for images to the automatic extraction of image features using deep learning techniques. This shift has led to significant improvements in model performance and has contributed to advancements in various domains. This report aims to introduce representative algorithms from different periods of computer vision development, providing a comprehensive understanding of the field's evolution and laying a solid foundation for future research.

2. Method

This report will use image recognition tasks as an example to sequentially elucidate representative algorithms in the field of computer vision in chronological order.

Section 2.1 will include an introduction to traditional image descriptors, Section 2.2 will cover an overview of deep learning methods, Section 2.3 will focus on the application of transfer learning in image recognition, and Section 3.4 will introduce current large-scale visual or multimodal models.

2.1. Traditional Image Descriptors

Firstly, an image descriptor is a simplified representation of an image, containing only the most important information about the image. By using image descriptors, we can obtain specific features of the image.

2.1.1 SIFT

SIFT [1] (Scale-invariant feature transform) is a classic feature point extraction and description algorithm in computer vision. This method was first published by David Lowe at the International Conference on Computer Vision (ICCV) in 1999 and was further refined and published in the International Journal of Computer Vision (IJCV) in 2004. SIFT features have several advantages:

1. They describe the local features of an image and maintain invariance to rotation (achieved by aligning the main orientation during feature descriptor generation), scale (ensured through the use of a difference of Gaussian pyramid), and brightness (ensured through feature normalization). They also exhibit robustness to changes in viewpoint, affine transformations, noise, and other variations.

2. They possess good uniqueness and rich information content, making them suitable for fast and accurate matching in large feature databases.

3. They exhibit multiplicity, meaning even a few objects can generate a large number of SIFT features.

4. When optimized, SIFT matching algorithms can achieve real-time performance.

5. They can easily be combined with other feature vectors.

The SIFT feature algorithm consists of four parts:

1) Scale-space extrema detection - Key point detection: To search for potential interest points (key points) that should be present at different scales and levels of blur, the algorithm first computes the difference of Gaussian images.

First, the original image is blurred at different levels, increasing the variance to create a Gaussian scale space. Each image in this scale space undergoes a series of isotropic downsampling, resulting in a Gaussian scale pyramid representing different spatial scales.

Next, within the same Gaussian scale pyramid, the differences mainly arise from varying degrees of Gaussian blur. To detect feature points that exist under different blur conditions, the algorithm uses the difference of Gaussian (DoG) instead of the Laplacian of Gaussian (LoG) due to its simpler computation. The calculation method is as follows:

$$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \quad (1)$$
$$= L(x, y, k\sigma) - L(x, y, \sigma) \quad (2)$$

where $G(x, y, k\sigma)$ is the Gaussian function, and $L(x, y, \sigma)$ represents the Gaussian scale space of the image.

In other words, subtracting two images with different levels of Gaussian blur in the Gaussian space results in the response image of the difference of Gaussians. (Here, k represents the Gaussian variance of the two adjacent images multiplied by k). By subtracting each pair of adjacent images within the same group, the difference of Gaussian pyramid is obtained.

Next, in order to find the extremum points, within the same pyramid, each pixel needs to be compared with all its image neighbors (physically adjacent pixels) and scale neighbors (adjacent Gaussian scale space) to determine if it is greater than/less than all its neighbors. If so, the point is considered an extremum point. In the Figure.1, all the green points are the ”neighbors” of the point x.

2) Key Point Localization: For each candidate key point, the final determination of its suitability, position, and scale is crucial. In SIFT feature extraction, key points are identified as extremal points of the pixel values in the DoG image neighborhood (including both the Gaussian scale neighborhood and the image neighborhood), as the DoG represents the gradient due to the differencing of pixel values.However, the local extremal points in the DoG are obtained through discrete space search and may not necessarily be true extremal points. Therefore, it is necessary to eliminate points that do not meet the conditions. This can be achieved by curve fitting using the scale-space DoG function to find extremal points, essentially removing points with highly asymmetric local curvature in the DoG. This step primarily eliminates feature points with contrast below a certain threshold and unstable edge response points (achieved through detecting principal curvatures).

3) Orientation Assignment for Key Points: One or multiple orientations are assigned to each key point based on the image’s gradient direction. After the previous steps, feature points that exist at different scales have been identified. In order to achieve rotation invariance, it is necessary to assign orientations to the feature points. For a particular feature point, its corresponding Gaussian scale image can be determined, and the gradient magnitude and orientation angles of the points within a 3x1.5 times the standard deviation range centered on this feature point can be computed:

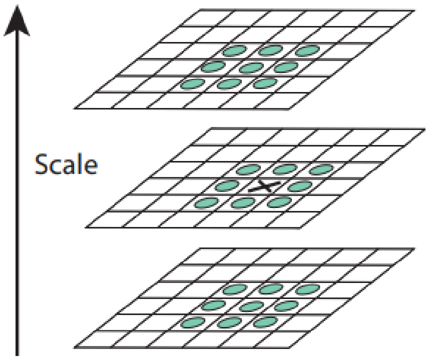


Figure 1. all the green points are the ”neighbors” of the point x.

$$\theta(x, y) = arctan\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (3)$$

The interval [0,360°] can be divided into 10 ranges, and the gradient histogram can be computed by accumulating the gradients within each range. The peak of the histogram represents the dominant orientation of the feature point (in practice, additional operations such as smoothing and interpolation should be applied).

In the gradient histogram, the presence of a bin value that accounts for 80% of the total energy can indicate an auxiliary direction for the feature point. Consequently, a feature point may detect multiple directions (or, equivalently, produce multiple feature points with the same coordinates and scale but different orientations).

Lowe stated in his paper that 15% of the keypoints have multiple orientations, and these points are crucial for the stability of matching.

Once the dominant orientation of the feature point is determined, each feature point can be characterized by three pieces of information (x, y, σ, θ), representing its position, scale, and orientation. This information defines a SIFT feature region, which is represented by its center (indicating the feature point position), radius (indicating the scale of the keypoint), and an arrow (indicating the dominant orientation). Feature points with multiple orientations can be duplicated, and the orientation values can be assigned to the duplicated feature points, resulting in multiple feature points with the same coordinates and scale but different orientations.

4)Key point descriptors: the gradient, scale, and position are all determined. Then they are transformed into a representation that can effectively handle image distortion, lighting, and other conditions.

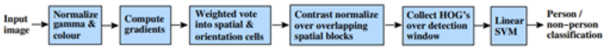

Input image
Normalize gamma & colour
Compute gradients
Weighted vote into spatial & orientation cells
Contrast normalize over overlapping spatial blocks
Collect HOG's over detection window
Linear SVM
Person / non–person classification

Figure 2. The process of the HOG algorithm in the original paper.

5)During the final detection, a database containing a large amount of trained image feature data will be used. The detection image will be compared with the feature data in the database, and recognition will occur when a sufficient number of matching features, at least three, are found. [2]

2.1.2 HOG

The HOG method [3], proposed in 2005, is now a popular feature descriptor technique in computer vision and image processing. It analyzes the distribution of edge directions within an object to describe its shape and appearance. The representation and shape of local targets can be well described by the density distribution of gradients or edge directions. The HOG method involves computing the gradient magnitude and direction for each pixel in the image, and then dividing the image into small cells. The original paper was aimed at effectively extracting pedestrian features. In Figure. 2

The specific operation steps are as follows: first, normalize the image pixels to reduce the impact of lighting changes on the results when obtaining image features. Then, calculate the gradient and direction of the image in blocks, store them in histograms, concatenate the histograms of different blocks, and then normalize again to obtain the complete feature vector of the image. At this point, the image features have been obtained and can be further processed. [4]

2.2. Deep Learning Method

In recent years, with the improvement of computing power, deep learning methods have also achieved great success. Unlike traditional feature extraction, deep learning methods automatically learn the representation of data through neural networks, eliminating the need for manual design of feature extractors. Deep learning models can be trained end-to-end with a large amount of data, automatically learning the feature representation in the data, thereby better capturing complex patterns and features in the data. This has led to significant progress in many fields, especially in computer vision, natural language processing, and speech recognition. In summary, traditional methods require manual design of feature extractors, while deep learning methods learn the representation of data through neural networks. Representative examples include CNN [5], RNN [6], and the Transformer [7], which has recently transitioned from the NLP field to the field of vision.

2.2.1 CNN

Convolutional Neural Network (CNN) is a type of deep learning model mainly used for processing and analyzing data with grid-like structures, such as images and videos. The design of CNN is inspired by the working principles of the human visual system, and it has achieved great success in the fields of image recognition, computer vision, and pattern recognition.

The basic structure of CNN includes convolutional layers, pooling layers, and fully connected layers. The convolutional layers extract features from the images through convolution operations, such as edges, textures, etc. The pooling layers are used to reduce the dimension of feature maps, decrease the number of parameters, while retaining important features. The fully connected layers are used to map the extracted features to different categories for the final classification or regression tasks.

The training of CNN is usually carried out through the backpropagation algorithm, using a large amount of labeled data for supervised learning. The emergence of CNN marks the formal rise and rapid development of deep learning algorithms. However, CNN also has certain limitations, such as a relatively large dependence on training samples, requiring a large amount of high-quality sample data support. In addition, the modeling effect for long-term dependencies is not ideal, resulting in more redundancy or poor processing effects when dealing with related entities. Therefore, in recent years, some variant CNN structures have also achieved great success, such as the introduction of residual blocks in residual networks, as well as the upcoming introduction of RNN, Transformer, and so on.

2.2.2 RNN

RNN is a type of neural network, similar to deep neural network, and generative adversarial network, and so on. RNN is very effective for data with sequential characteristics, as it can explore the temporal and semantic information in the data. In comparison to CNN, which does not incorporate context in training the model and instead trains the label of a particular sample separately, RNN emerged. The basic architecture of the RNN network model is shown in Figure 3.

The key difference between recurrent neural networks (RNNs) and CNNs is that RNNs process current and previous inputs in a time-sequential manner, using their output as a combined input for the next iteration. However, traditional RNNs have some limitations, including:

1. Long-term dependency problem: Traditional RNNs often encounter the issues of vanishing or exploding gradients when processing long sequential data, making it difficult to capture long-term dependency relationships.

2. Limited internal memory: Traditional RNNs have

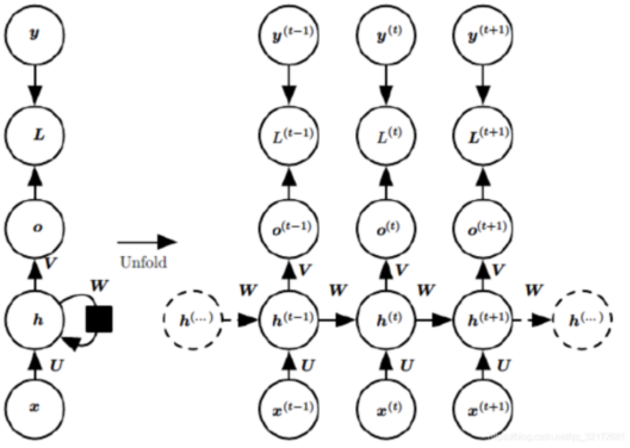


Figure 3. The basic architecture of the RNN network

limited internal memory capacity, making it challenging to effectively capture important information in long sequences.

3. Difficulty in parallelization: Due to the temporal dependency of RNNs, it is difficult to effectively parallelize data processing, which affects the efficiency of training and inference.

4. Sensitivity to input sequence length: Traditional RNNs are sensitive to the length of input sequences, often leading to decreased performance when the input sequence is long.

To overcome the limitations of traditional RNNs, researchers have proposed a series of improved models, such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), which have partially addressed some of the problems of traditional RNNs. In addition, there are also Attention-based models, such as Transformer, which have achieved great success in processing sequential data.

2.2.3 Transformer

Transformer originated from the 2017 paper ”Attention is all you need” by Google, initially used for machine translation with great success. Since then, the Transformer has excelled not only in the field of NLP but also demonstrated impressive performance in areas such as computer vision and recommendation systems. Particularly in 2020, it can be considered the prime time for the Transformer, as models based on it dominated various rankings in the field of computer vision. The following will introduce its principles and its applications in the field of computer vision.The basic architecture of the RNN network model is shown in Figure 4.

The Transformer consists mainly of an encoder, a decoder, and their connections. The encoder and decoder are

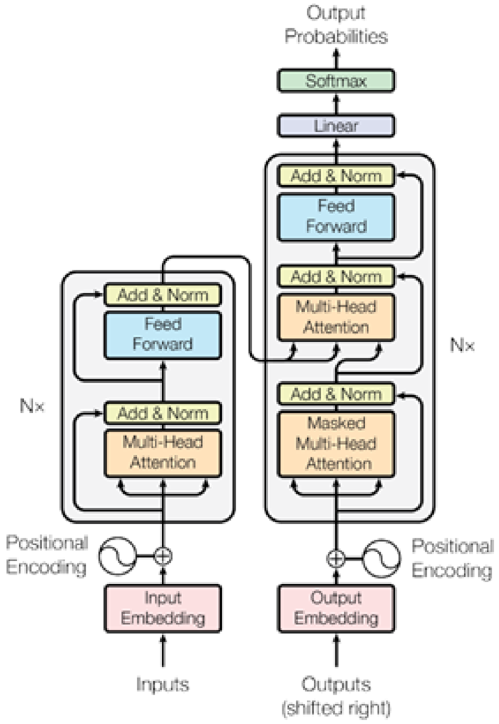


Figure 4. The basic architecture of the Transformer

responsible for transforming input and output into feature vectors, but the truly influential part of the transformer is its proposed Attention shown in Figure 5.

Stage 1: Calculate the similarity between Query and each Key to obtain a similarity score.

Stage 2: Convert the s scores into a probability distribution between [0,1] using softmax.

Stage 3: Use [a1, a2, a3...an] as the weight matrix to perform weighted summation on the Value to obtain the final Attention value.

$$A(Q,S)=\sum_{i=1}^{L_x} Sim(Q,Key_i)*V_i \tag{4}$$

In a typical Encoder-Decoder framework for tasks, the input Source and output Target content are different. For example, in English to Chinese machine translation, the Source is an English sentence, and the Target is the corresponding translated Chinese sentence. The Attention occurs between the elements of the Target and all elements of the Source. On the other hand, Self-Attention, as the name suggests, refers to the Attention not between the Target and Source, but within the elements of the Source or within the elements of the Target. It can also be understood as the Attention calculation that occurs in the special case where

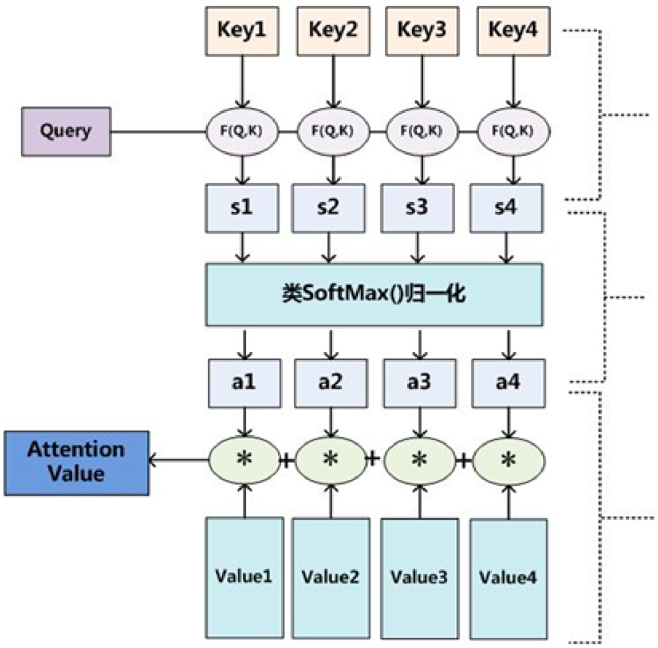


Figure 5. Basic principles of Attention.

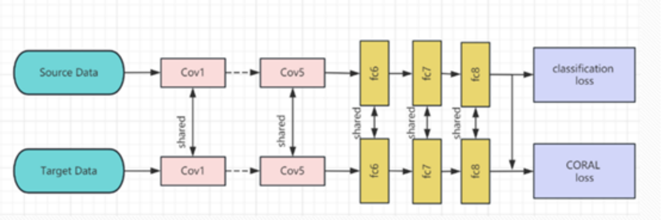


Figure 6. Basic principles of Deep Coral.

Target equals Source.

2.3. Transfer Learning

Transfer Learning is a machine learning method that leverages existing knowledge to assist in solving new problems. It can effectively address challenges such as limited data and difficult labeling, enhancing the model's generalization ability. Typically, this is achieved through pre-trained models. Pre-training involves training a model on a large dataset and then fine-tuning it on a new task.

In 2014, researchers introduced a neural network called DANN (Domain Adaptive Neural Network) at the Pacific Rim International Conference on Artificial Intelligence (PRICAI). DaNN falls under the category of Domain Adaptation. Domain adaptation is a crucial branch of transfer learning that aims to map data from different distributions in the source domain and target domain to the same feature space. The objective is to find a metric criterion to minimize the ”distance” in this space. Once a classifier is trained on the labeled source domain data, it can be directly applied to classify target domain data. Mainstream methods include Deep Coral and DANN shown in Figure 6 and Figure 7.

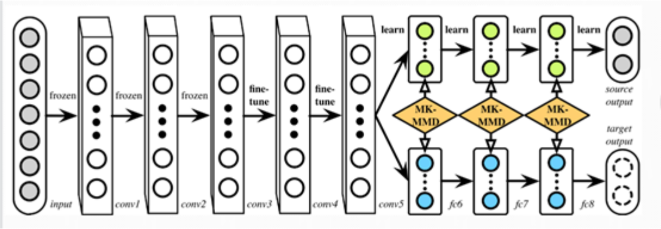


Figure 7. Basic principles of DANN.

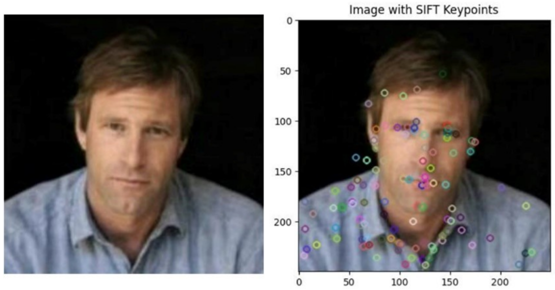


Figure 8. Each annotated circle in the image represents a feature point. These feature points are utilized during image classification to perform matching and selection.

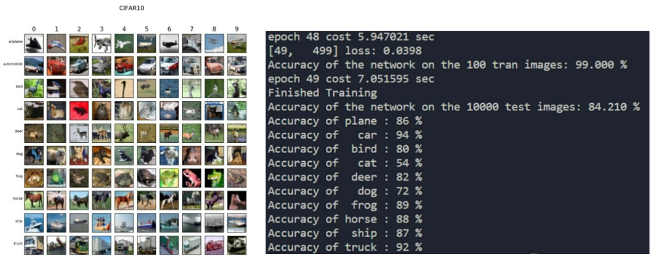
3. Experiment

In this chapter, various computer vision algorithms mentioned above were employed to achieve several effects. The examples are listed below.

Traditional image descriptors are generally characterized by strong theoretical foundations. However, due to their widespread use, they are often encapsulated into well-established methods in the field of computer vision. These methods can be easily invoked when needed. The image in Figure 8 provides a visual example of utilizing the SIFT method for feature description in computer vision.

For the CNN experiment in this study, the performance of a Convolutional Neural Network (CNN) was evaluated using the VGG-16 architecture as a representative example. The evaluation was conducted on the CIFAR-10 dataset for image classification.

The CIFAR-10 dataset in Figure 9 consists of 60,000 32x32 color images, distributed across 10 categories, with each category containing 6,000 images. Among these, 50,000 images are used for training, and 10,000 images are designated for testing. The dataset is divided into 5 training batches and 1 testing batch, each containing 10,000 images.



```
epoch 48 cost 5.947021 sec
[49,   499] loss: 0.0398
Accuracy of the network on the 100 tran images: 99.000 %
epoch 49 cost 7.051595 sec
Finished Training
Accuracy of the network on the 10000 test images: 84.210 %
Accuracy of plane : 86 %
Accuracy of   car : 94 %
Accuracy of  bird : 80 %
Accuracy of   cat : 54 %
Accuracy of  deer : 82 %
Accuracy of   dog : 72 %
Accuracy of  frog : 89 %
Accuracy of horse : 88 %
Accuracy of  ship : 87 %
Accuracy of truck : 92 %
```

Figure 9. The left image shows a partial display of the CIFAR-10 dataset, while the right image illustrates the results of the training process.

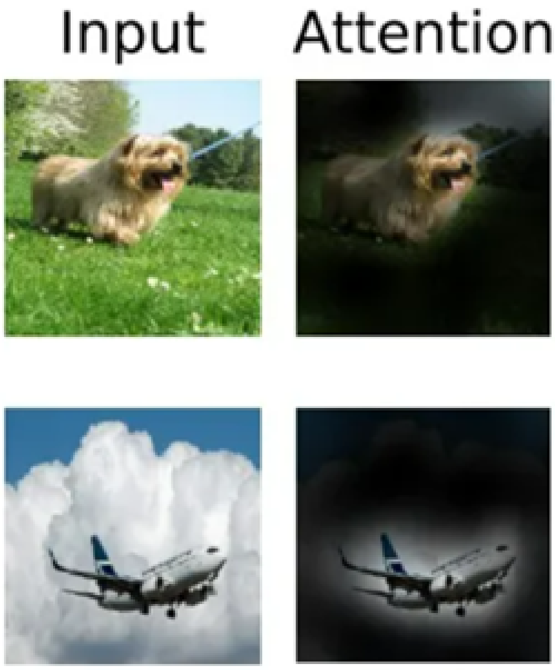


Figure 10. The visualization of Attention.

The testing batch comprises 1,000 randomly selected images from each category.The training batches contain the remaining images, with some training batches potentially having more images for a particular category than others. Each training batch includes 5,000 images from various categories. The classes are mutually exclusive, meaning that an image from one category will not appear in any other category. The image on the right side of Figure 9 depicts the training process and results of this experiment.

In the field of computer vision, CNNs have dominated, while in natural language processing (NLP), Transformers have become standard. In recent years, there has been a significant number of articles on the cross-application of Transformers to computer vision. Most of these approaches are based on two main ideas:

(1) Combining attention mechanisms with CNNs.

(2) Replacing certain CNN structures with attention mechanisms while keeping the overall architecture unchanged.

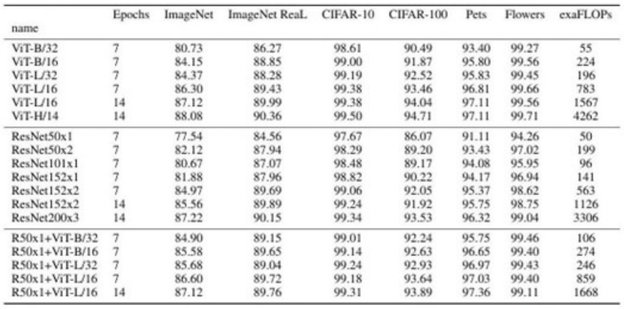
Figure 10 visualizes the Attention method, providing an explanation of the attention mechanism's capability to handle contextual relationships and capture essential information.

Due to limitations in the experimental environment, this implementation [8] may not be replicated independently. However, Figure 11 showcases the experimental results obtained by the authors of this paper.

On the experimental level, it was observed that ViT [8] (Vision Transformer) has a performance upper limit higher than ResNet. In other words, under conditions of a sufficiently large dataset, Attention s can fully replace CNNs (Convolutional Neural Networks).

The training for transfer learning was conducted locally, shown as Figure 12 and Table 1, and the experiment utilized two transfer learning algorithms: DANN [9] (Domain Adversarial Neural Network) and Deep Coral [10]. These algorithms were employed for image recognition, and the differences in experimental results were analyzed.

The experimental results indicate that the performance of both heterogeneous transfer learning approaches meets the



name	Epochs	ImageNet	ImageNet ReaL	CIFAR-10	CIFAR-100	Pets	Flowers	exaFLOPs
ViT-B/32	7	80.73	86.27	98.61	90.49	93.40	99.27	55
ViT-B/16	7	84.15	88.85	99.00	91.87	95.80	99.56	224
ViT-L/32	7	84.37	88.28	99.19	92.52	95.83	99.45	196
ViT-L/16	7	86.30	89.43	99.38	93.46	96.81	99.66	783
ViT-L/16	14	87.12	89.99	99.38	94.04	97.11	99.56	1567
ViT-H/14	14	88.08	90.36	99.50	94.71	97.11	99.71	4262
ResNet50x1	7	77.54	84.56	97.67	86.07	91.11	94.26	50
ResNet50x2	7	82.12	87.94	98.29	89.20	93.43	97.02	199
ResNet101x1	7	80.67	87.07	98.48	89.17	94.08	95.95	96
ResNet152x1	7	81.88	87.96	98.82	90.22	94.17	96.94	141
ResNet152x2	7	84.97	89.69	99.06	92.05	95.37	98.62	563
ResNet152x2	14	85.56	89.89	99.24	91.92	95.75	98.75	1126
ResNet200x3	14	87.22	90.15	99.34	93.53	96.32	99.04	3306
R50x1+ViT-B/32	7	84.90	89.15	99.01	92.24	95.75	99.46	106
R50x1+ViT-B/16	7	85.58	89.65	99.14	92.63	96.65	99.40	274
R50x1+ViT-L/32	7	85.68	89.04	99.24	92.93	96.97	99.43	246
R50x1+ViT-L/16	7	86.60	89.72	99.18	93.64	97.03	99.40	859
R50x1+ViT-L/16	14	87.12	89.76	99.31	93.89	97.36	99.11	1668

Figure 11. The testing results of the Transformer on different datasets and its comparison with other methods.

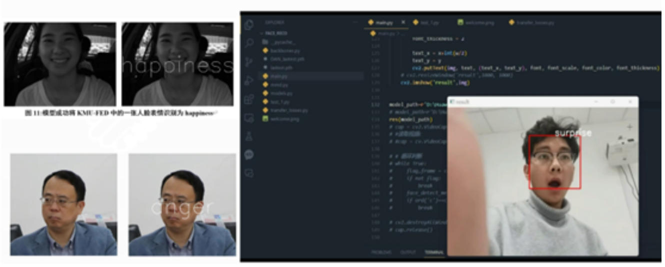


Figure 12. The figure represents experimental results on image recognition using transfer learning, specifically for facial expression recognition.

Method	Loss	Acc(%)
DANN	0.78	63.99
DeepCoral	0.69	62.56

Table 1. The result of transfer learning method.

basic requirements for discerning facial expressions. However, the accuracy of the training results is not overly precise. The possible reasons for this could be attributed to the following four points:

1. Significant Discrepancy Between Source and Target Domains: The images in the source domain, MMA FACIAL EXPRESSION, consist of European and American portraits, while the target domain comprises images of Asians. Due to the substantial facial differences between Europeans/Americans and Asians, this results in a lower accuracy.

2. Insufficient Training Iterations, Incomplete Model Fitting: Limited by training time and resources, we conducted training for 50 iterations. It is reasonable to believe that the lower test results may be due to incomplete model fitting resulting from the limited training iterations.

3. Improper Parameter Adjustment: Parameters of the transfer learning model may require more detailed adjustments and optimization to adapt to the target domain's data. In our case, we directly used default optimal parameters from other transfer learning tasks as hyperparameters for this task. This improper parameter setting may have contributed to the suboptimal results.

For the two algorithms implemented in this task, the accuracy of DANN is superior to Deepcoral. This further proves that directly performing feature transformation and re-fitting on the data source and target domain is more effective than simply reducing cross-domain losses. It also provides valuable insights for our future algorithm design considerations.

4. Conclusion

This paper aims to present a report and review of fundamental concepts in the field of computer vision. It includes the application of traditional image descriptors such as HoG, SIFT, etc., in tasks like image recognition. Additionally, the paper explores the application of deep learning methods, including CNN, RNN, Transformer, etc., as well as transfer learning in computer vision tasks such as image recognition. The paper emphasizes the importance of both traditional and deep learning approaches in the field of image recognition. Traditional methods like HoG and SIFT are utilized for image description, while deep learning methods like CNN, RNN, and Transformer demonstrate strong performance in computer vision tasks. Furthermore, the paper focuses on the application of transfer learning in computer vision, highlighting its potential to enhance model performance and generalization. Overall, the paper provides readers with a comprehensive understanding of key concepts in the field of computer vision and discusses the advantages and limitations of different approaches in addressing image recognition challenges.

The experimental code for the paper has been uploaded to GitHub. For detailed information, please visit the website: https://github.com/Michael8023/cv_report.

References

[1] A. Witkin. Scale-space filtering: A new approach to multi-scale description. 9:150–153, 1984. 1

[2] Crystal. A summary of commonly used traditional image processing methods, 2023. 3

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. 1:886–893 vol. 1, 2005. 3

[4] MaWB. Introduction to image feature engineering: Hog feature descriptor, 2023. 3

[5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3

[6] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 3

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016. 6

[10] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 443–450. Springer, 2016. 6