基於 Transformer 的離線手寫簽名驗證模型結構研究

Research on the Structure of Offline Handwritten Signature Verification Models Based on Transformer

姓　名 Name	：	王茗琛
學　號 StudentNo.	：	2230025907
學　院 Faculty	：	創新工程學院
課　程 Program	：	智能技術碩士
專　業 Major	：	智能技術
指導教師 Supervisor	：	劉新
日　期 Date	：	2025/05/27

摘 要

離線手寫簽名驗證是生物特徵技術的應用, 其根據用戶提供的手寫簽名与存儲的参考簽名對比以驗證用戶身份, 被廣泛用於安全認證, 金融交易等安全領域. 學術研究中, 學者們定義了作者獨立和作者依賴任務以評估離綫手寫簽名驗證模型的泛化和判斷性能.

深度學習的視覺模型在離綫手寫簽名驗證領域中取得了顯著進展, 但是模型對高像素的手寫簽名圖像特徵學習能力具有一定的缺陷, 并且注意力機制的 Transformer 具有訓練過程模型收斂速度慢的缺點. 為了在一定程度上解決該問題, 本工作基於 Transformer 提出 OSVTF 端到端模型結構, 在組合特徵的基礎上增加多尺度融合特徵, 并且對部分子網絡進行了加速訓練收斂的優化.

在實驗部分, OSVTF 模型在 BHSig-B 80/20 數據集的作者獨立和作者依賴任務中 EER 為 4.54% 和 2.13%, 在 BHSig-H 100/60 數據集的 EER 為 3.90% 和 2.68%, 達到了日常生產使用的水準. 在消融實驗部分, OSVTF 在 CEDAR 50/5 數據集的作者獨立任務中 EER 為 4.75%, 證實加速訓練收斂優化的有效性. 整體實驗結果證實在跨數據集的場景下, OSVTF 相較過去的離綫簽名驗證模型在性能上取得了一定程度的提升, 在此基礎上具有更優秀手寫簽名圖像特徵學習能力和更快的訓練收斂速度.

關鍵詞：離線手寫簽名驗證; 多尺度融合; 端到端; OSVTF.

Abstract

Offline handwritten signature verification is an application of biometric technology, which is widely used in security verification, financial transactions, and other security domains to verify the identity of a user based on the comparison of the user-supplied handwritten signature with the stored reference signature. In academic research, writer-independent and writer-dependent tasks have been defined to evaluate the generalization and judgmental performance of offline handwritten signature verification models.

Deep learning visual models have made significant progress in the field of offline handwritten signature verification, but the models have some shortcomings in their ability to learn the features of high pixel handwritten signature images, and the Transformer with attention mechanism has the disadvantage of slow convergence of the model during the training process. In order to solve the problem to a certain extent, this work proposes an end-to-end model structure of OSVTF based on Transformer, which adds multi-scale fusion features on the basis of combined features, and optimizes some sub-networks to accelerate the training convergence.

In the experimental part, the OSVTF model achieves EERs of 4.54% and 2.13% for writer-independent and writer-dependent tasks on the BHSig-B 80/20 dataset, and 3.90% and 2.68% on the BHSig-H 100/60 dataset, which is up to the level of daily production use. In the ablation experiment part, OSVTF has an EER of 4.75% in the writer-independent task on the CEDAR 50/5 dataset, which confirms the effectiveness of the accelerated training convergence optimization. The overall experimental results confirm that OSVTF achieves a certain degree of performance improvement over previous offline signature verification models in cross-dataset scenarios, based on which it has better handwritten signature image feature learning capability and faster training convergence speed.

KeyWords： Offline handwriting signature verification; multi-scale fusion; end-to-end; OSVTF.

Table of Contents

List of Figures

List of Tables

List of Symbols

$\mathcal{B}$	Backbone
$\mathcal{F}$	FPN Fusion
$\mathcal{H}$	Holistic Encoder
$\mathcal{C}$	Conv-Module
$\mathcal{P}$	Contrast based Part Decoder
H	Image Height
W	Image width
C	Num of image channel
$\mathbf{I}$	Input image
$\mathbf{F}$	Feature map
$\boldsymbol{x}$	Feature vector
f	Flatten feature vector
$\tilde{\mathbf{W}}$	2D convolutional kernel weight
$\tilde{\mathbf{b}}$	2D convolutional kernel bias
$\tilde{\mathbf{P}}$	2D convolutional padding value
$\tilde{\mathbf{S}}$	2D convolutional stride
$\mathbf{z}$	Conv2D output feature map
$\hat{\mathbf{z}}$	Normalized feature map
γ	Learnable bias of normalization
β	Learnable weight of normalization
s_i	Upsampling horizontal axis magnification factor
s_i	Upsampling ordinate magnification factor
N	Num of feature length
P	swing window size
S	swing window step size
D	Num of feature dimensions
$f^{\mathcal{H}_{cls}}$	Learnable weight of Holistic Encoder
$f^{\mathcal{P}_{cls}}$	Learnable weight of Contrast based Part Decoder
$\overline{\boldsymbol{x}}$	Intermediate layer feature vector
$\overline{f}$	Intermediate layer flatten feature vector
$\boldsymbol{X}$	MHSA query/key/value matrix
d_q	Feature dimensions of query matrix in MHSA
d_k	Feature dimensions of key, value matrix in MHSA
$\mathbf{T}$	Input projection linear weight of MHSA
$\mathbf{L}$	Output projection linear weight of MHSA
$\mathcal{E}$	Transformer encoder layer output feature vector
$\mathbf{W}_1$	First linear weight of Feed Forward Network

$\mathbf{b}_1$	First linear bias of Feed Forward Network
$\mathbf{W}_2$	Second linear weight of Feed Forward Network
$\mathbf{b}_2$	Second linear bias of Feed Forward Network
$\mathcal{G}$	GeLU function
h	Height of feature map in 2D convolution
w	Width of feature map in 2D convolution
M	Num of head in multi-head attention
$d_{\mathcal{P}}$	Feature dimensions of cross multi-head attention
$d'_{\mathcal{P}}$	Feature dimensions of cross multi-head attention after linear projection
$\mathbf{T}^{\mathcal{P}}$	Weight of cross multi-head attention
$\mathcal{D}$	distance of a pair feature
m	Forged sample loss boundary value
n	Genuine sample loss boundary value
σ	Sigmoid function
$\overline{K}$	Genuine sample loss scaling factors
$\overline{V}$	Forged sample loss scaling factors
α_1	Genuine sample margin value of loss function
α_2	Forged sample margin value of loss function
λ_1	Weight values of multi-scale feature loss
λ_2	Weight values of encoder feature loss
λ_3	Weight values of convolutional module feature loss
λ_4	Weight values of decoder feature loss
ξ	Weight value of sparisity loss
$\mathcal{L}$	Loss function

List of Abbreviations

OSVTF	Offline signature verification transformer
WI	Writer-independent
WD	Writer-dependent
CNN	Convolutional neural network
ResNet	Residual network
BN	Batch normalization
ReLU	Rectified linear unit function
FPN	Feature pyramid network
ViT	Vision transformer
SVM	Support vector machine
MLP	Multilayer perception
Conv2D	2D convolution
GeLU	Gaussian error linear unit function
Add	Residual linking
LN	Layer Normalization
MHSA	Multi-head self attention
MHA	Multi-head attention
Attn	Scaled dot product attention
FFN	Feed forward network
Conv-Module	Convolutional module
Up	Up-sample
GAP	Global average pooling
CMHA	Cross multi-head attention
FAR	False acceptance rate
FRR	False rejection rate
EER	Equal error rate
ACC	Accuracy

Chapter 1 Introduction

1.1 Research Background

Biometrics technology is a technology that identifies or verifies individuals based on their physiological characteristics such as fingerprints, face, iris, or behavioral characteristics such as voiceprints and handwritten signatures. This technology is widely used in security authentication, financial transactions, access control systems and other security areas [19] for personal identification or verification. In the identification scenario, the system will identify the user profile already in the system database based on the physiological or behavioral characteristics provided by the user. This scenario is applicable to fingerprint, iris personal identification, etc. Secondly, in the verification scenario, the user needs to provide the system with verified identity and characteristic information, and the system will judge whether the current user is the declared user based on the stored characteristic information and the information currently provided by the user, which is applicable to the scenarios of declaring personal identity and providing personal characteristic information, such as the unlocking of smart phones and international border crossing.

Handwritten signature is a more important individual behavioral characteristic in daily life, and as the main characteristic for verifying personal identity in legal, financial, administrative and other fields, because it cannot be intruded during the collection process, handwritten signature is also regarded as one of the main characteristics of many technologies for verifying personal identity. Handwritten signatures produce different writing styles, such as regular and cursive, depending on the individual’s writing habits. Even with the passage of time, the personal writing style may change, and the signature (defined as Query) provided by the user after a certain period of time may differ from the reference signature (defined as Reference) previously entered into the system. As a result, there will be some difficulties in comparing the verification of the user’s handwritten signature, for example, there will be differences in the angle of the bending strokes, the length of the straight line, and the hook at the end of each character on the personal signature. As mentioned above, the handwritten signature may be slightly different in the end due to different writing habits of individuals. Therefore, in order to obtain the identity authority of a certain user, some people will maliciously forge handwritten signatures in order to pass the security and privacy checking system, or even practice for a long period of time in order to achieve a handwritten signature of a similar style as that of a certain user. As a result, academics have launched a series of studies on handwritten signature verification, hoping to design a relevant model to help relevant staff to more efficiently complete the process of more repetitive handwritten signature verification work, so that more staff to participate in the core work.

In academia, handwritten signature verification is categorized into two types, offline and online, depending on the data collection route. The collection process of offline handwritten signatures is obtained after the user's writing process on paper; online handwritten signatures are collected by using a digitizing station, and the collected handwritten signature images may be affected by the equipment, such as the position of the pen, tilt, and pressure, etc. [8]. In addition to the different data collection routes are categorized into different types of handwritten signature verification, scholars have defined two types of branching tasks, Writer-Independent (WI) and Writer-Dependent (WD) to evaluate the model algorithms. The WI task is needed to input a pair of handwritten signature image to the model structure, based on the features extracted from the reference signature and the query signature so as to determine whether the query signature is forged or not. the WD task adds the author id to the input of the WI task, and uses the corresponding author classifier according to the author id in order to determine whether the query signature is forged or not. These two branching tasks, although both output the label category of query signatures (genuine or forged) at the end, can reflect the different performance of the model to different degrees. The WI task focuses on learning the global characteristics of the signature population, does not need to learn a separate writer classifier for each user, and has the excellent extensibility of not requiring re-training for new users, and at the same time can reflect the generalization and discriminative ability of the model. The WI task focuses on learning the global features of the signature group, does not need to learn an writer classifier for each user individually, has excellent scalability without re-training for new users, and can reflect the model's generalization and discriminative ability; the WD task additionally trains an writer classifier for each user individually, which can take into account the high accuracy of different styles of signatures of individual writers that are still considered to be real signatures, but needs to collect a sufficiently large number of samples for each user, with a high cost of training and a weak generalization ability. Thus the WI and WD tasks can reflect the overall performance of the model to different degrees, and the model performance is comprehensively evaluated according to the evaluation indexes of the two tasks.

With the development of the level of technology, the field of artificial intelligence has been able to generate fake pictures based on the sample images and keywords provided by the user, and it is difficult for normal people to identify the real and fake with the naked eye. Therefore, there will be part of the artificial generation of forged signature behavior, so as to achieve the fake to pass the verification of the security system, resulting in personal privacy, property invasion, theft and other dangerous consequences. Relying on manual power to verify will have certain social risks and judgment errors, so the study of offline handwritten signature verification, to a certain extent, can reduce the risk of forged signatures through the verification of the relevant algorithmic models to help minimize the error of manual verification, so as to better ensure personal privacy and property security.

1.2 Research Motivation and Importance

This work focuses on the offline handwritten signature verification task, which is essentially a derivative branch of an image categorization task. Unlike the traditional image categorization task, the offline handwritten signature verification task requires the input of a reference and query pair of signature image samples, and the WD task needs to additionally input an writer id in order to train an writer-independent classifier. The offline handwritten signature verification model will extract relevant image features based on the input reference and query signature images, and compare the features to output genuine or forged category labels. Thus the task is challenging and innovative, the design of the model or algorithm architecture not only needs to consider the criticality of the model to extract a pair of image features, the model architecture of the isotropic dual-stream will directly affect the criticality of the features extracted from the reference signature and query-signature images, and once the extracted features don't have so important information, it will lead to a drastic decrease in the model's ability to make judgments. At the same time, due to the definition of the WI and WD tasks and the real-life application scenarios, it is required to have a high accuracy rate and generalization ability to prevent personal privacy and property infringement, but also to enhance the demand for model performance, which makes the design and experimentation of the offline handwritten signature verification model a very challenging image task.

In the design of offline handwritten signature verification model, it is divided into two parts: feature extractor and writer classifier for WD task. In the past researches of scholars, most of them manually design the image feature extraction method based on the data cluster features such as sample distribution, and take the traditional machine learning method to determine whether the signature is forged or not [19], such as taking the distance between the features or SVM to determine whether the signature is forged or not. But this manually designed features have the defects of specific data clusters, it must require the data set of the writer's signature style uniformity, subtle differences will be as outliers leading to the traditional machine learning methods to determine the accuracy of the decline in the accuracy of the work, but the actual production work is required to have these certain differences in the tolerance of the degree of the work, so this method of taking the artificial design of the extracted features is gradually phased out. Scholars hope to have an image feature extractor that focuses more on a certain part of the image and does not need to intervene many times. With the rapid development of deep learning in the past six years, convolutional neural network (CNN) with shared parameters for local field of view operations has achieved good results in traditional image classification tasks such as MNIST, ImageNet, and compared with the traditional machine learning methods, the accuracy and generalization ability of CNN is even better than that of traditional machine learning methods. The accuracy and generalization ability of CNNs have been further verified in comparison with traditional machine learning methods. As a result, scholars in the field of handwritten signature verification have introduced CNN as a feature extractor, and experiments have proved that the accuracy and generalization ability of this approach has made a great breakthrough compared with previous methods [18].

Even though CNN's ability to extract image features is outstanding, its core idea of convolutional kernel operation has a certain local field of view reinforcement learning ability, which can focus on the local features of the image; however, for handwritten signatures not only need to pay attention to the degree of the corners of the font strokes, but also need to pay attention to the overall style of the signature image, fonts, and other factors, which may result in the case of the signature of the same writer in a different location, so CNN as a feature extractor still has some defects. With the development of the field of natural language processing, the appearance of Transformer [3] with global feature learning attracted the attention of scholars, and then scholars in the field of image introduced the Encoder-Decoder architecture of Transformer to experiment on the image classification task [5], and the experiments proved that the performance of this approach is better than CNN's feature extractor in the case of convergence of model parameters. It is proved that this approach has better performance than CNN's feature extractor in the case of model parameter convergence, but the conditions for training to achieve model parameter convergence are more demanding, because Transformer's attention mechanism needs to learn the whole image, whereas CNN adopts the way of sharing parameters to learn the local features of the image, so that model convergence can be achieved faster, but the generalization ability is poorer in practice, and we need to keep fine-tuning the dataset in order to achieve higher model performance. However, in practice, the generalization ability is poor and the dataset needs to be constantly fine-tuned to achieve higher model performance. As a result, CNN is derived as Backbone to extract multi-channel feature maps, which are flattened as feature vectors and entered into the Transformer for global attention feature operation [10], so as to achieve the multi-channel feature maps in the absence of overall information, and then strengthen the characteristics of the overall information through the Transformer's attention mechanism, which is proved by experiments to be more effective in learning image features and modeling. Experiments have proved that this approach can learn image features more effectively, and the model thus achieves better accuracy and generalization ability. In summary, the CNN+Transformer style model architecture has become the mainstream framework in the image field in recent years, and it has also been introduced in offline handwritten signature verification tasks and experiments have proved that this style of model architecture has good model performance [10], so the offline handwritten signature verification model based on Transformer proposed in this work is also a CNN+Transformer style model architecture. Transformer style model architecture, in line with the development trend of this field in deep learning in recent years, and on this basis, multi-scale fusion features are added, in order to meet the growing requirements of high-definition image resolution at the same time, to learn the signature image font stroke corners and other features at multiple scales, to make up for the shortcomings of the multi-channel feature maps in terms of the scale after the CNN. In the final output classification stage, unlike the traditional image classification task, the model proposed in this work will collect the features from the previous modules in the final output features, and perform overall stitching for category label prediction or model training, which has been proven to be effective [40]. The challenge of the research is how to effectively use multi-scale fusion features for category prediction and model training, this way of training and prediction using the overall model features can better train the model weights of each part in a holistic manner, and the model training to the convergence stage will surpass most of the previous deep learning model architectures, which is a novel idea for the related fields to integrate the feature approach of convolutional neural networks and attention mechanisms. For related fields, it provides a novel idea to integrate the features of convolutional neural network and attention mechanism, which provides a certain design idea for the subsequent more concise fusion module; on the contrary, this approach will increase the computational reasoning pressure of the equipment and the cost of model training, and it needs to be proved by various experiments that this approach is effective.

1.3 Research Objects

The Offline Signature Verification TransFormer (OSVTF) structure proposed in this work, as a whole, is a model designed based on the idea of siamese networks, as shown in Fig. 1.1.

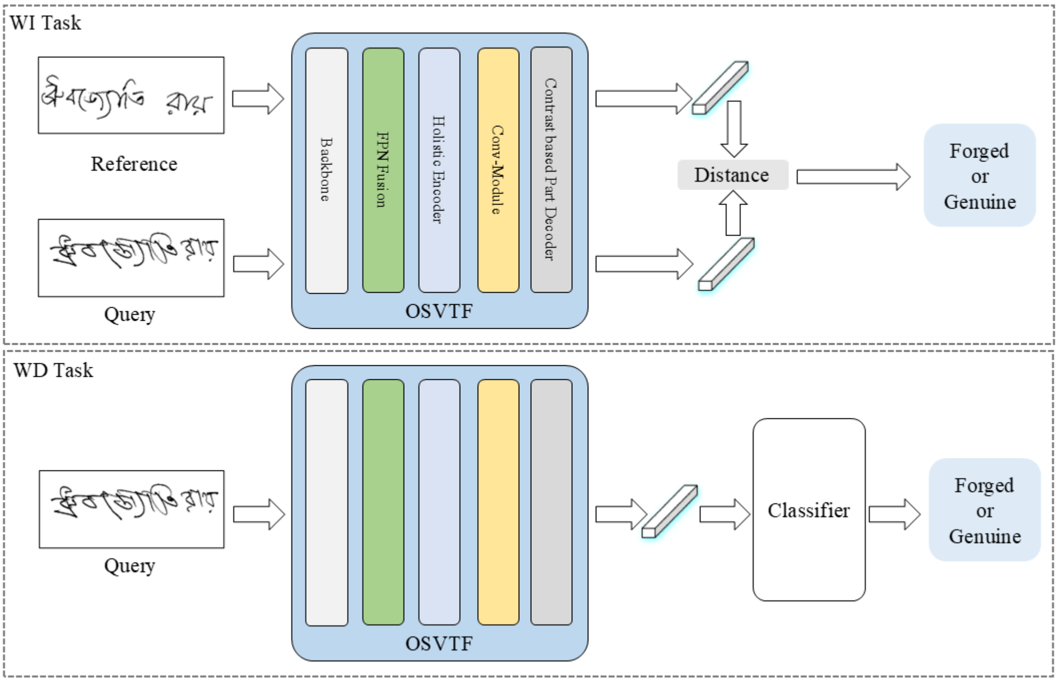


Fig. 1.1: OSVTF overview

The core idea of the siamese network is two sub-networks with shared weights that accept two inputs and output two feature vectors, and subsequently calculate the similarity using a distance function [28]. In the WI and WD tasks of offline handwritten signature verification, a pair of images of the inputreference and query signatures are input, so the overall architecture is referred to TransOSV [40], based on which the backbone and FPN Fusion modules are added, and the Encoder, Conv-Module, and Decoder sections are subjected to a The Encoder, Conv-Module and Decoder parts are optimized and adjusted in a series of ways, which are more conducive to the change and learning of the features of each part, and the experimental part is needed to confirm that these modules can improve the model's ability of judging forged signatures and generalization ability in different scenarios.

In the overall research plan and experiments, the two general directions of

WI and WD are still adopted to evaluate the performance and quality of the model as a whole, based on which the multi-channel feature maps of CNN and the attention mechanism of Transformer will be visualized, and we need to pay attention to whether these modules are able to better grasp some of the important feature information of the image in the image feature learning. In addition, there are many ways of fusion of multi-scale features, such as Mask R-CNN [16] takes the feature pyramid network style (FPN-Style) [35], this kind of fusion includes, but is not limited to, mean accumulation, direct summation, and splicing, so the follow-up will be to train the model of these fusion methods with the control variables in order to get the best performance of fusion of multi-scale feature maps way. In addition, the previous twin network approach is to calculate the similarity based on the distance function in order to complete the related tasks, whether this approach can still be effective after the addition of multi-scale fusion features requires some verification, this work will refer to the traditional image classification task of the CNN processing classification approach, the addition of a global average pooling layer as a classifier to classify the features label prediction, in order to determine whether it would be outperform the previous distance similarity judgment.

In summary, this work will be divided into three parts of the research phase: 1. Initially, the OSVTF model architecture will be trained in the model cycle to verify whether the optimized and adjusted scheme of multi-scale features and models can be improved on the original architecture; 2. For the multi-scale fusion method of the FPN Fusion module, a small fine-tuning training will be taken on the basis of the first phase of the control variable method for the model, and the best multi-scale feature fusion method will be selected. The best multi-scale feature fusion method; 3. For the WD task, a global average pooling classifier is added in the final classifier stage, which is compared with the previous method of distance similarity prediction for offline handwritten signature verification, so as to judge the advantages and disadvantages of the two classifiers, and a better classifier is adopted for experiments in the WD task to verify whether it works.

Chapter 2 Literature Review

In Biometrics technology, handwritten signatures are more important personal identity verification features, and the core challenge of its verification algorithms or models lies in the high intraclass variation of signatures (natural variations of signatures of the same person) and the low interclass variation (similarity between forged signatures and real signatures), and the models and algorithms will be evaluated based on these two directions. The handwritten signature verification task, on the other hand, is categorized into offline and online verification depending on how the image data is processed. In this work, focus on the offline handwritten signature verification task with static images, which is different from online verification in that the user's signature is dynamically accepted on the digitizing device, i.e., the handwriting data is changing in real time, and all the information from the beginning to the end of the process of writing will be collected; and the offline verification is completely through the static image data in order to carry out the user's verification, which does not set any limitations on the digitizing device, and it only needs to be done through the This method does not set any limitations on the digitizing device, and only requires the author to write on a form, thus greatly reducing the workload in the data collection process.

The image data collected by offline handwritten signature needs to be preprocessed to a certain extent for model or algorithm validation, and good data preprocessing techniques will to a certain extent improve the system validation accuracy. In order to improve the quality of signature images, scholars have adopted the image processing of converting RGB images to GRAY single-channel images and smoothing pixels to remove noise [20]. This processing can well reduce the unnecessary features of handwritten signature images collected on white paper, such as the value of some channel pixel points in the RGB three-channel image is 255, while the value of pixel points with handwritten handwriting is often very small. However, this processing method has the defect that the value of the pixel points in the part with handwriting features is very small while the value of the pixel points in the other blank parts is very large, so the subsequent scholars processed these images with black and white inversion, and it was verified that this processing method can get a better verification accuracy rate [18].

In the early stage of offline signature verification in the 1990s-2000s period, scholars based on geometric features to distinguish the authenticity of the signature image sample pairs [26], by manually extracting the image aspect ratio, stroke length and other global features and inflection points, curvature, and other local features for matching recognition. This approach laid the initial feature extraction template for offline signature verification, where similarity calculation is performed after manual feature extraction. In 1996, Lee et al. proposed template matching for offline signature verification, which aligns the signature trajectories by dynamic time regularization (DTW) to determine whether the signature is forged or not [17]. With the development of statistical theory, two schools of work emerged in the 2000s-2010s period: feature engineering dominated by a series of feature design and classifier design. In 2011, Vargas et al. mentioned for the first time the introduction of feature engineering into feature extraction for offline signature verification by combining texture engineering (LBP, HOG) with structural features (gradient direction distribution). These feature engineering will lead to a drastic increase in the number of signature image features, thus requiring scholars to start adopting machine learning approaches in order to act as classifiers, e.g., Edson et al. proposed an offline signature verification system in 2005 using Hidden Markov Models [9], where the HMM model focuses on modeling the variation of the allowable changes in signatures, i.e., treating each pixel of the image as a word vector, in order to better capture the relationship between the strokes relationships; Malik et al. in 2013 used SVM as a classifier to accomplish handwritten signature verification with higher dimensional feature counts, SVM is one of the most commonly used classifiers for binary classification tasks among academics, and it can also deal with high feature dimensions, so most of the follow-ups have taken SVM in the classifier stage. Even though traditional machine learning methods have improved the performance of early signature verification systems, they still have the defect of relying greatly on manual feature extraction, and manually designing a large number of feature extraction steps will lead to too large a difference in the individual feature parts, which cannot ensure that the merged feature vectors can be statistically significant, and manually designing the features is difficult to capture complex patterns, such as pen pressure and local texture. Moreover, the models trained by this method have insufficient generalization ability, are poorly adapted to cross-dataset and cross-linguistic scenarios, and are extremely costly in terms of computational time [39].

With the development of computer hardware, offline handwritten signatures are gradually transitioning to deep learning. The early stage of deep learning was a development process dominated by convolutional neural networks, and in 2012 Krizhevsky A. et al. proposed AlexNet [1] to win the ImageNet competition's image classification task far ahead of the second-place participant using traditional machine learning algorithms, which made deep learning represented by convolutional neural networks gradually become the mainstream method for image tasks. Subsequently, academics introduced a deeper convolutional neural network, VGG [14], which is an architecture that adds more convolutional layers on top of AlexNet to extract more channels of feature maps. However, this computation of stacked convolutional layers makes the feature map size smaller while leading to the loss of features at each stage, in order to solve this problem, in 2016 K. He et al. proposed ResNet with residual connectivity [15]. A residual connection is added between the convolutional layers of each size feature map to accumulate the feature maps from the original input model layer to the output feature maps. This way in supervised learning convolutional neural network, after weight sharing convolutional kernel operation, can somewhat compensate for the original image feature information, and this residual connection becomes the mainstream way of retaining local features in subsequent deep models. The high-quality performance of convolutional neural networks on image classification tasks has led to offline handwritten signature verification to start resorting to convolutional neural networks in order to replace the earlier step of manually designing features. In 2017, L. G. Hafemann et al. used convolutional neural networks to extract signature image features for the first time for verification [18], adopting the convolutional partial operation to output feature maps as signature image features, and designing two fully-connected layers in the output stage to share the signature image features to output the writer ID and signature category. This approach combines both WI and WD tasks, satisfies both outputs and is shown to significantly improve the accuracy of offline signature verification. This approach and its reliance on model training skills, the training process and its difficulty in achieving parameter convergence results, but also provides ideas for subsequent handwritten signature verification tasks using deep learning methods. In 2017, Y. Hafemann et al. proposed that using pre-trained CNNs (e.g., VGG) to extract global and local features of signature images and feeding the features into SVMs for classification can significantly improve generalization across datasets [18] with end-to-end optimization. Similarly M. Diaz et al. in 2019 trained CNNs via migration learning combined with SVM as a joint classifier to achieve low error rates on the GPDS dataset [12]. This CNN+SVM approach successfully reduces the error rate of the signature verification task model to a level unmatched by traditional machine learning, which are followed by this deep learning model as a feature extractor and SVM as a joint classifier in order to perform signature verification. And according to the image size of the dataset, S. Dey et al. proposed SigNet [32] in 2020, which fuses multi-scale CNN features with SVM, adopting different scale feature maps outputted from multiple convolutional layers in CNN as signature image features, and optimizing classification boundaries by combining with SVM, and experiments have proved that the multi-scale features will greatly improve the robustness of the overall model, and have a very good model across datasets. The experiments proved that multi-scale features will greatly improve the robustness of the overall model and have very good model performance across data sets. However, CNN has difficulty in capturing the global contextual information of signature images, such as the long-distance dependence between strokes, in the learning ability of signature image features.

With the development of the natural language processing field of deep learning, A. Vaswani et al. proposed the Transformer with Encoder-Decoder architecture based on the attention mechanism in 2017 [3], the attention mechanism can have the ability of global feature awareness for word vectors, which can better capture the contextual information in machine translation tasks with variable sentence lengths. This kind of machine translation task reflects its excellent global generalization ability, and the excellent global context-awareness ability of the architecture is reflected in the translation task across multiple different languages. In view of Transformer's global feature-awareness capability, A. Dosovitskiy et al. first used Transformer's attention mechanism for image classification tasks in 2021, proposing Vision Transformer (ViT) [5] without convolutional operations, which chunks flattens and maps the input image as word vectors to input into Transformer Encoder for attention feature computation, and finally MLP is performed to generate image category probabilities using the feature vectors output from Encoder. This way of image feature processing method that maps the image element as token after mapping, got the recognition accuracy ahead of CNN on Transformer architecture, and reflected the powerful global generalization ability of Transformer on validation and test set, so all kinds of tasks in the field of image gradually began to take the attention mechanism for secondary processing of image features. In 2022, X. Wang et al. for the first time, ViT for offline signature verification, to solve the above CNN is difficult to capture the global context information of the signature image, the use of self-attention mechanism to model the global structure of the signature, to improve the differentiation of complex forged signatures [38], in the GPDS-960 and CEDAR [30] datasets compared to the same period of time the model's Equal Error Rate decreased by 12%. In 2023 Y. He et al. also proposed TransOSV [40] based on ViT, which is an architecture that provides an end-to-end Transformer-based offline signature verification system that combines both global semantic information and local discrepancy information, and more efficiently utilizes Transformer for local and global feature modeling. It introduces a multi-layer feature combination mechanism that splices features from different modules into the final signature representation to achieve multi-angle detection of forged signatures; meanwhile, Focal Contrast Loss is designed based on the Focal Loss [28] of the comparison task and the handling of the sample imbalance Double-Margin Loss [37] to solve the above problems while Enhance the learning of hard samples in the public dataset of offline signature task, and also incorporate the comparison learning of different module features. And unlike the previous twin networks for offline signature verification, the part decoder is proposed, which locates and focuses on the sensitive parts of forged signatures by comparing the token-level attention of reference and query, and introduces the weight of the attention mechanism into the part features of the decoder, which enhances the model's perception of the part location of forged signatures. However, TransOSV still has defects, it adopts a multi-layer feature combination mechanism to detect signature image pairs of features from different perspectives, but it does not have a strong generalization ability in the case of different signature image scales, and the decoder part does not have a good abstract expression ability for token-level features.

In summary, this work will add backbone to extract multi-scale feature maps on the basis of TransOSV architecture, and adopt FPN Fusion to fuse multi-scale features in order to enhance the learning of multi-scale features by the model architecture. At the same time, in order to better abstract expression of multi-scale fusion features in the subsequent modules, the Conv-Module is designed with downsampling structure, and the input mapping as well as the multi-head adjustment of cross-attention are added in the part decoder, which enhances the model's ability of abstract expression of image features, and is able to strengthen the attention to the forged signatures after the learning of global contextual attention features in the Encoder localized sensitive regions. On the WD task, a global average pooling layer [22] will be provided for signature category prediction as per the traditional image classification task due to the adoption of a large number of convolutional output feature maps and a multi-layer feature combination mechanism.

Chapter 3 Methodology

3.1 Overview

In this work, It will define the label categories forged as 1 $(y = 1)$ and genuine as 0 $(y = 0)$ for handwritten signatures, and the input reference signature image is defined as $R \in \mathbb{R}^{H\times W\times 1}$ and the query signature image is defined as $Q \in \mathbb{R}^{H\times W\times 1}$. The architecture of the proposed Offline Signature Verification TransFormer (OSVTF) model for multi-scale feature fusion is shown in Fig. 3.1, which consists of five parts, namely, Backbone ($\mathcal{B}$), FPN Fusion ($\mathcal{F}$), Holistic Encoder ($\mathcal{H}$), and Conv-Module ($\mathcal{C}$), Contrast based Part Decoder ($\mathcal{P}$) five parts.

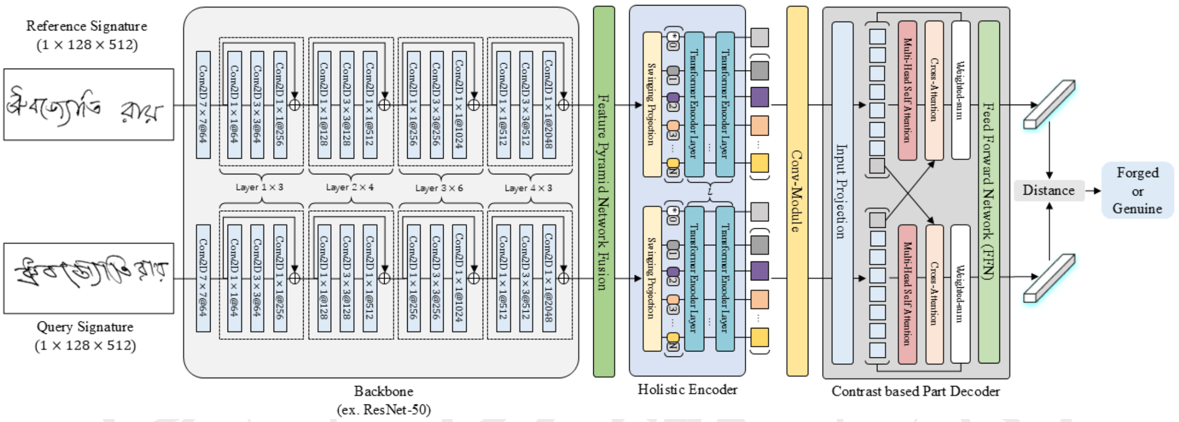


Fig. 3.1: OSVTF structure

In the model, it will input a pair of signature image samples $\mathbf{I}_r$ and $\mathbf{I}_q$, the initial input image goes through Backbone with shared weights (ResNet-50 [15] is used as an example in Fig. 3.1) to get the multi-scale feature map set $\mathcal{B}(\mathbf{I}_r), \mathcal{B}(\mathbf{I}_q)$, and then the different scale feature maps are fused through FPN Fusion module, and the output of multi-scale fusion feature maps $\mathbf{F}_r^{\mathcal{F}}, \mathbf{F}_q^{\mathcal{F}}$. Then it enters the Holistic Encoder with shared weights to get the holistic flat features $f_r^{\mathcal{H}}, f_q^{\mathcal{H}}$ and patch embeddings $\boldsymbol{x}_r^{\mathcal{H}} = [x_r^{\mathcal{H}_1}, \cdots, x_r^{\mathcal{H}_N}]$ and $\boldsymbol{x}_q^{\mathcal{H}} = [x_q^{\mathcal{H}_1}, \cdots, x_q^{\mathcal{H}_N}]$. To further integrate the feature maps, the above patch embeddings are reshaped to a 2D shape, and its outputs $\mathbf{F}_r^{\mathcal{C}}, \mathbf{F}_q^{\mathcal{C}}$ are obtained after the Convolutional Module (Conv-Module) with shared weights, and they will go directly to the Contrast based Part Decoder to obtain the cross-attention contrast part flat features $f_r^{\mathcal{P}}, f_q^{\mathcal{P}}$. In the final prediction stage, Global Average Pooling (GAP) operation is performed on $\mathbf{F}_r^{\mathcal{F}}, \mathbf{F}_q^{\mathcal{F}}$ and $\mathbf{F}_r^{\mathcal{C}}, \mathbf{F}_q^{\mathcal{C}}$ to obtain the multiscale fusion flat features $f_r^{\mathcal{F}}, f_q^{\mathcal{F}}$ and convolution flat features $f_r^{\mathcal{C}}, f_q^{\mathcal{C}}$. The total feature vector $f_r = [f_r^{\mathcal{F}}, f_r^{\mathcal{H}}, f_r^{\mathcal{C}}, f_r^{\mathcal{P}}]$ and $f_q = [f_q^{\mathcal{F}}, f_q^{\mathcal{H}}, f_q^{\mathcal{C}}, f_q^{\mathcal{P}}]$ of the OSVTF are obtained by splicing all the flat features, and the judgment of whether $\mathbf{I}_q$ is forged or not will be made based on f_r and f_q. Next, Backbone, FPN Fusion, Holistic Encoder, Conv-Module and Contrast based Part Decoder structures will be analyzed in depth.

3.2 Backbone

In the Backbone section an structure of CNN will be adopted that discards Average Global Pooling and Fully Connected Layers in the output section.Most of the CNN architecture consists of a convolutional layer in the input section and four defined convolutional layers, as shown in Fig. 3.2 for example in ResNet-50 [15], where each layer consists of a number of bottleneck blocks.

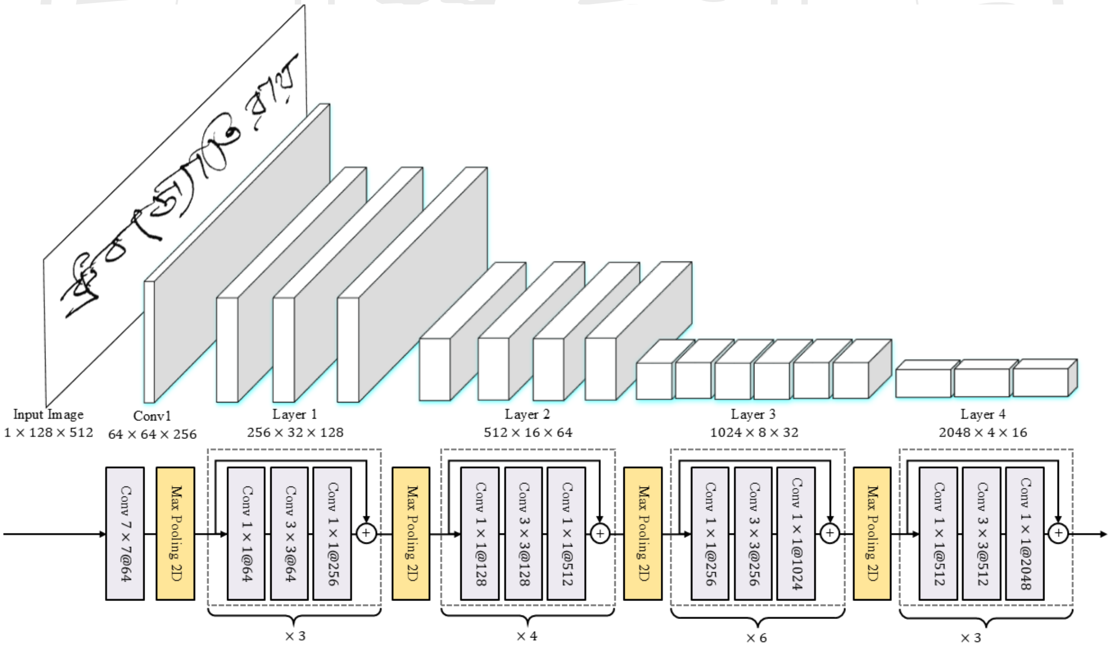


Fig. 3.2: Backbone structure

A single bottleneck block consists of two convolutional 2D layers with convolutional kernels of $(1 \times 1 \rightarrow 3 \times 3 \rightarrow 1 \times 1)$, and this arrangement mainly has the following purposes:

1. Reduce the feature dimensions to achieve the purpose of improving the

computational efficiency;

2. The intermediate 3×3 convolution extracts the local spatial features such as strokes, edges, etc., in the lower dimensional space has the ability to maintain the sensory field while reducing the redundancy, and this reduces the number of channels before expanding to the input, and then the number of channels is expanded to the input, and the number of channels is reduced. This first reduces the number of expression channels and then expands to the input expression channels to find better feature learning parameters in the bottomleneck, which has a better feature abstraction and expression ability;

3. In the actual operation, it can be paired with the identity skip connection to quickly propagate the gradient to avoid gradient disappearance, and at the same time, such a design makes the number of intermediate parameters and the burden of computation greatly reduced, and it can stack more layers. At the same time, this design can greatly reduce the number of intermediate parameters and computational burden, and can stack more layers, such as ResNet-101, to promote the training of deeper networks;

4. This modular design has the advantage of easy fine-tuning of the convolutional parameters, and the number of intermediate channels can be adjusted according to the environment, so as to control the balance of performance and efficiency. In addition, ResNet series network introduces residual connection, which solves the problem of missing features after deep convolutional operation of the image. Residual connection is introduced in each bottomleneck block, which accumulates the feature map of the input part with the output of the bottomleneck, which can reduce the problem of feature loss after a certain number of convolutional layer operations to a certain extent.

Assume that the input feature map $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ of the convolutional layer, the output feature map $\boldsymbol{z} \in \mathbb{R}^{H' \times W' \times C'}$, the convolutional kernel size of the convolutional 2D layer $K \times K$ (the convolutional kernel weight $\tilde{\mathbf{W}} \in \mathbb{R}^{K \times K \times C \times C'}$, the padding $\tilde{P}$, the step size $\tilde{S}$, and the bias term $\tilde{b}$. For the c'-th output channel of the feature map, the pixel point with spatial location (i, j) is calculated as eq. 3.1.

$$\boldsymbol{z}(i,j,c') = \sum_{c=1}^{C'} \sum_{u=1}^{K} \sum_{v=1}^{K} = \tilde{\mathbf{W}}_{u,v,c,c'} \cdot \mathbf{I}_{i\cdot\tilde{S}+u-\tilde{P},j\cdot\tilde{S}+v-\tilde{P},c} + \tilde{\mathbf{b}}_{c'} \tag{3.1}$$

In addition, in each bottleneck block, except for the last 1×1 convolutional 2D layer which is followed only by the BatchNorm (BN) [31] operation, each convolutional 2D layer will be followed immediately by the BN and ReLU activation function [36] operation (Conv2D→BN→ReLU), and the spatial location (i, j, c') after the BN and ReLU activation function is the pixel point values are as eq. 3.2.

$$\hat{\boldsymbol{z}}(i,j,c') = \max\left(0, \gamma \frac{\boldsymbol{z}(i,j,c') - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta\right) \tag{3.2}$$

Where μ, σ^2 are the mean and variance of the feature statistics of the c' channel plane in the batch multi-channel feature maps output from the convolutional 2D layer, and γ, β are the learnable scaling and translation parameters. This shows that the idea of the BN operation is to normalize a certain batch of data, which has the effect of improving the convergence speed when training the model parameters. In each bottomleneck block operation, a cumulative residual operation is performed on the feature maps after three convolutional 2D operations, i.e., the feature maps input to the bottomleneck block are added to the feature maps output from the three convolutional 2D operations, which are then passed to the next bottomleneck block. At the same time, each layer of the backbone operation will be followed by a downsampling operation as a way to reduce the redundant features on the multi-channel feature map, which will be taken as Max Pooling 2D in order to equate the downsampling operation [15]. The above residual linking approach can ensure that the feature map retains certain original image features after the convolution operation, so the ResNet family of networks is most suitable as the backbone of the depth model to extract the image multi-channel feature map.

The backbone is taken mainly to extract the image multi-scale feature maps, so the different scale feature maps of each layer except the input convolutional layer are included in the final output as eq. 3.3.

$$\mathcal{B}(\mathbf{I}) = \{\mathbf{F}^{\mathcal{B}(1)}, \mathbf{F}^{\mathcal{B}(2)}, \mathbf{F}^{\mathcal{B}(3)}, \mathbf{F}^{\mathcal{B}(4)}\} \tag{3.3}$$

where the multi-channel feature map $\mathbf{F}^{\mathcal{B}}(i) \in \mathbb{R}^{\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times \frac{C}{2^{i-1}}}, (i = 1, 2, 3, 4)$ for each scale, H, W, C denote the width, height and the number of channels of the output feature maps of layer 1, and $C = 256$ when the backbone is ResNet-50), and each layer of the feature map has a twofold relationship between the channels and dimensions.

3.3 FPN Fusion

In order to be able to effectively utilize the multi-scale feature maps extracted by backbone, the Feature Pyramid Network Fusion (FPN Fusion) module is used in order to fuse feature maps at different scales, which adopts the model architecture of FPN-Style [35], as shown in Figure 3.3.

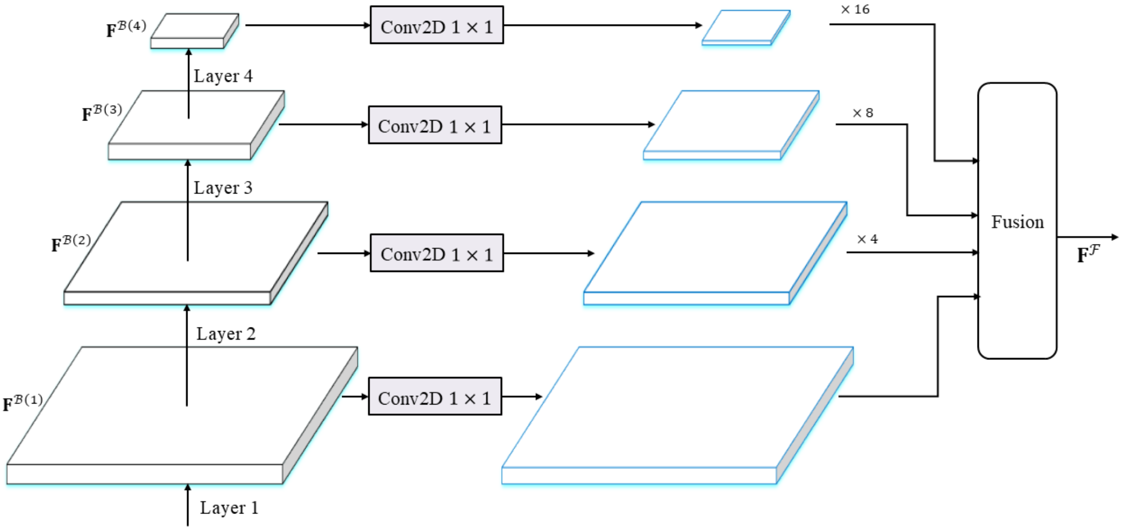

$\mathbf{F}^{\mathcal{B}(4)}$
Conv2D 1 × 1
× 16
Layer 4
$\mathbf{F}^{\mathcal{B}(3)}$
Conv2D 1 × 1
× 8
Layer 3
$\mathbf{F}^{\mathcal{B}(2)}$
Conv2D 1 × 1
× 4
Fusion
$\mathbf{F}^{\mathcal{F}}$
Layer 2
$\mathbf{F}^{\mathcal{B}(1)}$
Conv2D 1 × 1
Layer 1

Fig. 3.3: FPN Fusion module structure

Where the bottom-up process is the process of backbone inference to obtain different scales of feature maps. In bottom-down, for each layer of feature maps, a convolutional 2D layer with a convolutional kernel of 1×1 will be taken for mapping, which maps the feature maps with different number of channels at each scale to the same feature dimensions, and achieves the operation of dimensionality reduction to a certain extent, so as to accelerate the speed of model inference. At the same time, $\mathbf{F}^{\mathcal{B}(2)}, \mathbf{F}^{\mathcal{B}(3)}, \mathbf{F}^{\mathcal{B}(4)}$ are up-sampled to increase the size to the same size as $\mathbf{F}^{\mathcal{B}(1)}$, and then fused according to the fusion method in order to get the multi-scale fusion feature map. Bilinear interpolation [25] is adopted in the up-sampling process, for the lth layer feature map up-sampling output $\mathbf{F}^{\mathcal{B}(l)}$, the value of spatial location (i', j', c) is calculated as eq. 3.4.

$$\begin{aligned}\text{Up}(\mathbf{F}^{\mathcal{B}(l)})(i', j', c) = {} & (1-\delta_i)(1-\delta_j) \cdot \mathbf{F}^{\mathcal{B}(l)}(i, j, c) \\ & + \delta_i(1-\delta_j) \cdot \mathbf{F}^{\mathcal{B}(l)}(i+1, j, c) \\ & + (1-\delta_i)\delta_j \cdot \mathbf{F}^{\mathcal{B}(l)}(i, j+1, c) \\ & + \delta_i\delta_j \cdot \mathbf{F}^{\mathcal{B}(l)}(i+1, j+1, c)\end{aligned} \tag{3.4}$$

where $\delta_i = \frac{i'}{s_i} - \lfloor \frac{i'}{s_i} \rfloor, \delta_j = \frac{j'}{s_j} - \lfloor \frac{j'}{s_j} \rfloor$ denote fractional portions of the value in the direction of the width and the height, s_i, s_j denote the magnification ($s_i = s_j = 2$), and $\lfloor \cdot \rfloor$ denotes the value of the integer part. This up-sampling method can well preserve the original feature ground structure while introducing a smooth transition as shown in Fig. 3.4.

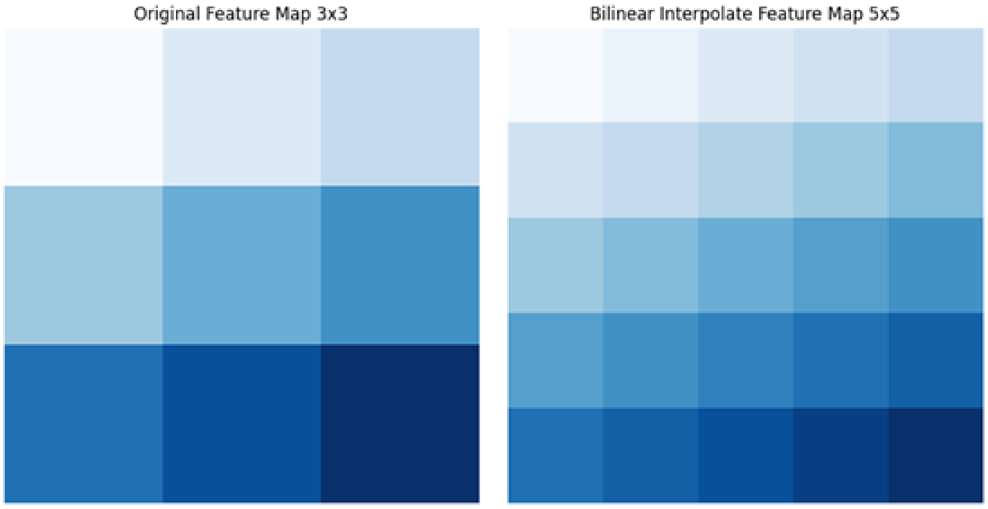


Fig. 3.4: Bilinear interpolation example

Each up-sampled pixel point will be weighted according to the original feature map pixel points, which not only retains the structural scale of the original feature map, but at the same time makes the transition between pixel points smoother, and to a certain extent, it can reflect the effect of different scales on the image feature points, for example, the features of a certain stroke will be enlarged.

After each scale feature map is sampled on the ground by bilinear interpolation, it will be fused according to three ways: accumulation, weighted average, and channel splicing. The first way is the traditional FPN-Style approach, which directly performs the accumulation operation on the downscaled multi-scale feature maps directly, similar to the residual link mentioned above. The design idea of the second way is to be able to ensure that the signature images of different scenarios can have a certain learning ability, by setting the proportion of weights to be able to pay more attention to the features of a certain scale, and the default is to take all equal weights. The third way is to splice in the channel dimension, followed by a convolutional kernel 1×1 convolutional 2D operation, this method can maximize the retention of all scales of features, but may lead to the model computational overhead is too large, resulting in slower convergence of the model.

After fusing the multi-scale feature map, it will finally go through a convolutional 2D layer with a convolutional kernel 3×3 to smooth out the checkerboard artifacts brought about by the up-sampling process, and finally output the multi-scale fused feature map $\mathbf{F}^{\mathcal{F}} = \mathcal{F}(\mathcal{B}(\mathbf{I})) \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C}$.

3.4 Holistic Encoder

The Holistic Encoder is designed with reference to Vision Transformer (ViT) [5]. In the input part, swing embeddings [41] are adopted in the input stage instead of the previous chunking operation, where chunking and mapping are performed on the input multiscale fusion feature maps according to the set window size $P \times P$ and step size S , the patch embeddings are obtained and then flattened to be used as feature vectors $[\overline{x}^{\mathcal{H}_1}, \overline{x}^{\mathcal{H}_2}, \cdots, \overline{x}^{\mathcal{H}_N}] \in \mathbb{R}^{N \times D}$, where D denotes the number of mapped feature dimensions, and the vector length, N, is computed as eq. 3.5.

$$N = \lfloor \frac{H/4 - P + S}{S} \rfloor \times \lfloor \frac{W/4 - P + S}{S} \rfloor \tag{3.5}$$

The Holistic Encoder structure is shown in Fig. 3.5.

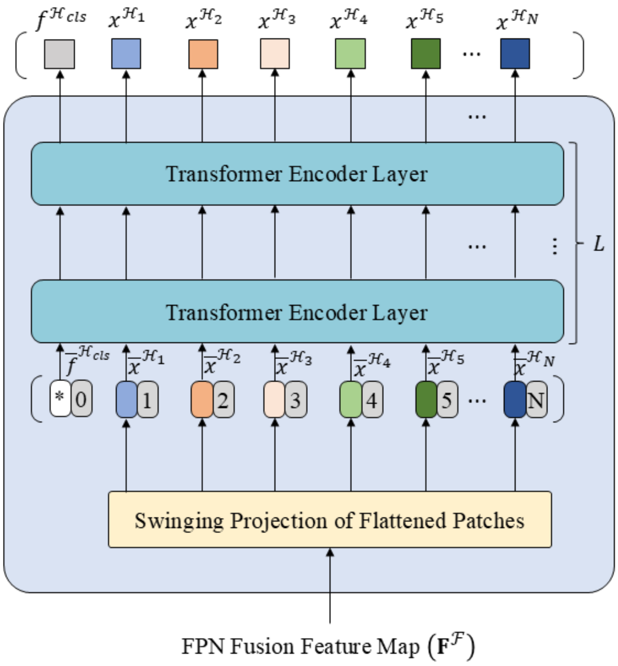
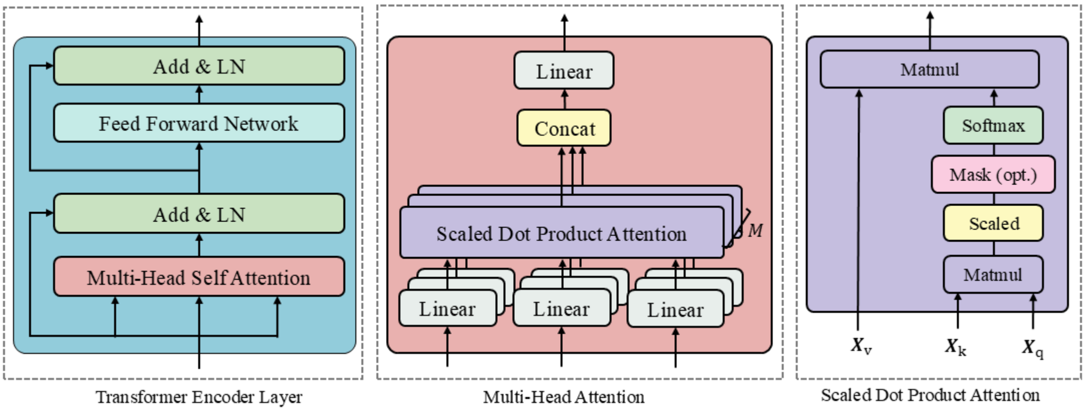

$f^{\mathcal{H}_{cls}}$
$x^{\mathcal{H}_1}$
$x^{\mathcal{H}_2}$
$x^{\mathcal{H}_3}$
$x^{\mathcal{H}_4}$
$x^{\mathcal{H}_5}$
$x^{\mathcal{H}_N}$
Transformer Encoder Layer
L
Transformer Encoder Layer
$\overline{f}^{\mathcal{H}_{cls}}$
$\overline{x}^{\mathcal{H}_1}$
$\overline{x}^{\mathcal{H}_2}$
$\overline{x}^{\mathcal{H}_3}$
$\overline{x}^{\mathcal{H}_4}$
$\overline{x}^{\mathcal{H}_5}$
$\overline{x}^{\mathcal{H}_N}$
* 0
1
2
3
4
5
N
Swinging Projection of Flattened Patches
FPN Fusion Feature Map ($\mathbf{F}^{\mathcal{F}}$)

Fig. 3.5: Holistic Encoder structure

Splicing a learnable weight $\overline{f}^{\mathcal{H}_{cls}} \in \mathbb{R}^D$ defined as a class token in swing embeddings, i.e., the feature vector of the input Transformer Encoder Layer is $\overline{\boldsymbol{x}}^{\mathcal{H}} = [\overline{f}^{\mathcal{H}_{cls}}, \overline{x}^{\mathcal{H}_1}, \overline{x}^{\mathcal{H}_2}, \cdots, \overline{x}^{\mathcal{H}_N}] \in \mathbb{R}^{(N+1)\times D}$. Due to the flattening operation performed on the feature map, positional embeddings [3] need to be accumulated on the feature vectors as a way to make the feature vectors contain information about their positions on the multiscale fused feature map.

After swing embeddings, the feature vector $\overline{\boldsymbol{x}}^{\mathcal{H}}$ enters the Transformer Encoder Layer for attention feature computation. Where Transformer Encoder Layer is formed by L stacks as shown in Fig. 3.6.



Transformer Encoder Layer Multi-Head Attention Scaled Dot Product Attention

Fig. 3.6: Transformer Encoder Layer structure

Each Transformer Encoder Layer consists of a Multi-Head Self Attention → Add & LN and Feed Forward Network → Add & LN, where Add & LN denotes residual linking and LayerNormalization (LN) [13]. The input feature vectors first enter Multi-Head Self Attention (MHSA) in order to perform the computation of the attention mechanism, which is essentially Multi-Head Attention (MHA) except that the sources before the mapping of $\boldsymbol{X}_{\mathrm{q}} \in \mathbb{R}^{N \times d_q}$ and $\boldsymbol{X}_{\mathrm{k}}, \boldsymbol{X}_{\mathrm{v}} \in \mathbb{R}^{N \times d_k}$ in the operations are The same, i.e., the input feature vectors will be simultaneously entered into the linear mapping layer of each head for dimensionality reduction, thus generating the query vector $\boldsymbol{X}_{\mathrm{q}}$, the key vector $\boldsymbol{X}_{\mathrm{k}}$, the value vector X_v, and $d_q = d_k$. where the attention mechanism is divided into multiple head computations in order to conserve the computational resources, and the feature vectors will be entered into the three linear layers of the M heads in order to be dimensionality reduced for mapping, thus achieving the attention computation in parallel, and the final splice and linear layer mapping will input vectors with the same feature dimensions, so the output vector shape of MHSA is the same as the input vector shape. The attention feature vectors of each head in MHSA are computed by taking scaled dot product attention [3] with the following eq. 3.6.

$$\mathrm{Attn}(\boldsymbol{X}_{\mathrm{v}}, \boldsymbol{X}_{\mathrm{k}}, \boldsymbol{X}_{\mathrm{v}}) = \mathrm{Softmax}\left(\frac{\boldsymbol{X}_{\mathrm{q}}\boldsymbol{X}_{\mathrm{k}}^{\mathrm{T}}}{\sqrt{d_k}}\right) \cdot \boldsymbol{X}_{\mathrm{v}} \tag{3.6}$$

In this case, there is a soft query relationship between $\boldsymbol{X}_{\mathrm{q}}, \boldsymbol{X}_{\mathrm{k}}$ and $\boldsymbol{X}_{\mathrm{v}}$. Instead of selecting a specific key-value pair for each query, all key-values are weighted and averaged, and the weights are computed by the similarity between $\boldsymbol{X}_{\mathrm{q}}$ and $\boldsymbol{X}_{\mathrm{k}}$. each feature token of $\boldsymbol{X}_{\mathrm{q}}$ establishes a relevance distribution, i.e., a Softmax [23] of attention weights, based on which the values of $\boldsymbol{X}_{\mathrm{v}}$ are weighted and fused. The advantage of this mechanism is that it can focus on multiple relevant tokens at the same time, which enables the model to more fully model the contextual information, for example, the (i, j) position of attention weight indicates the weight of i-th position of $\boldsymbol{X}_{\mathrm{q}}$ on the j-th position of $\boldsymbol{X}_{\mathrm{k}}$, the larger the value of this weight, the stronger the contextual relationship of the query at the i-th position to the key at j-th position, and conversely the smaller the value, the relationship is very small. According to this attention weight, multiplying $\boldsymbol{X}_{\mathrm{v}}$

again will further amplify the features in the place where the attention weight of its own vector is high, and the attention information will be included between tokens, so as to achieve that each token owns its own global information, and can more effectively utilize the important features and ignore the useless features, and will more effectively pay attention to the part of the brush strokes in the image of the handwritten signature, ignoring the blank part. The resulting MHSA output of a single Transformer Encoder Layer is as eq. 3.7.

$$\mathrm{MHSA}(\overline{\boldsymbol{x}}^{\mathcal{H}}, \mathbf{T}, \mathbf{L}) = \begin{bmatrix} \mathrm{Attn}(\overline{\boldsymbol{x}}^{\mathcal{H}}\mathbf{T}_{1,1}, \overline{\boldsymbol{x}}^{\mathcal{H}}\mathbf{T}_{1,2}, \overline{\boldsymbol{x}}^{\mathcal{H}}\mathbf{T}_{1,3}) \\ \mathrm{Attn}(\overline{\boldsymbol{x}}^{\mathcal{H}}\mathbf{T}_{2,1}, \overline{\boldsymbol{x}}^{\mathcal{H}}\mathbf{T}_{2,2}, \overline{\boldsymbol{x}}^{\mathcal{H}}\mathbf{T}_{2,3}) \\ \cdots \\ \mathrm{Attn}(\overline{\boldsymbol{x}}^{\mathcal{H}}\mathbf{T}_{m,1}, \overline{\boldsymbol{x}}^{\mathcal{H}}\mathbf{T}_{m,2}, \overline{\boldsymbol{x}}^{\mathcal{H}}\mathbf{T}_{m,3}) \end{bmatrix}^{\mathrm{T}} \cdot \mathbf{L} \tag{3.7}$$

where $\mathbf{T}_l \in \mathbb{R}^{M \times 3 \times d_k \times d'_k}$ denotes the linear mapping layer weights in the input phase of the MHSA, $\mathbf{L} \in \mathbb{R}^{d_k \times d'_k}$ denotes the weights of the final linear mapping layer in the MHSA, and d'_k denotes the number of feature dimensions of each head $\boldsymbol{X}_{\mathrm{q}}$, $\boldsymbol{X}_{\mathrm{k}}$ and $\boldsymbol{X}_{\mathrm{v}}$, and thus must satisfy $d_k = M \times d'_k$.

Matrix multiplication with larger dimensions is performed in the MHSA section, and in order to reduce the risk of gradient explosion or vanishing, and also to reduce the missing information of the original feature vectors, a residual link and an LN layer follow immediately after the MHSA.The main role of the LN layer is to normalize the feature vectors at each position, thus speeding up the convergence of the model as well as improving the model stability. Unlike the mean and variance of the BN, the mean and variance of the LN are derived from the current feature dimensions on which they are calculated as eq. 3.8.

$$\begin{aligned} \mu &= \frac{1}{d_k} \sum_{d}^{d_k} \mathrm{MHSA}(\overline{\boldsymbol{x}}^{\mathcal{H}}, \mathbf{T}, \mathbf{L})_d \\ \sigma^2 &= \frac{1}{d_k} \sum_{d}^{d_k} \Big(\mathrm{MHSA}(\overline{\boldsymbol{x}}^{\mathcal{H}}, \mathbf{T}, \mathbf{L})_d - \mu \Big)^2 \end{aligned} \tag{3.8}$$

where $\mathrm{MHSA}(\overline{\boldsymbol{x}}^{\mathcal{H}}, \mathbf{T}, \mathbf{L})_d$ denotes the dth feature dimension of the MHSA output feature vector. The subsequent normalization operation is the same as in the BN layer, and the entire MHSA→Add & LN output is computed as eq. 3.9.

$$\hat{\text{MHSA}}(\overline{\boldsymbol{x}}^{\mathcal{H}}, \mathbf{T}, \mathbf{L}) = \gamma \cdot \left(\frac{\text{MHSA}(\overline{\boldsymbol{x}}^{\mathcal{H}}, \mathbf{T}, \mathbf{L}) - \mu}{\sqrt{\sigma^2 + \epsilon}} + \mathcal{E}_{l-1} \right) + \beta \tag{3.9}$$

Where $\mathcal{E}_{l-1}$ denotes the output of the previous Transformer Encoder Layer, if $l = 1$, then $\mathcal{E}_{l-1} = \overline{\boldsymbol{x}}^{\mathcal{H}}$. After the attention computation, Feed Forward Network (FFN) is introduced to enhance the overall model to model the complex patterns [11], and the whole FFN is computed as eq. 3.10:

$$\text{FFN}(\hat{\text{MHSA}}(\overline{\boldsymbol{x}}^{\mathcal{H}}, \mathbf{T}, \mathbf{L})) = \mathcal{G}(\hat{\text{MHSA}}(\overline{\boldsymbol{x}}^{\mathcal{H}}, \mathbf{T}, \mathbf{L})\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{3.10}$$

Where $\mathbf{W}_1 \in \mathbb{R}^{D \times d_{ff}}, b_1 \in \mathbb{R}^{d_{ff}}$ denotes the weight and bias of the first linear layer in the FFN, and $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times D}$, $\mathbf{b}_2 \in \mathbb{R}^D$ denotes the weight and bias of the second prior layer in the FFN, $\mathcal{G}$ denotes the GeLU function [7]. the GeLU function and the ReLU function act similarly as the activation function that performs the nonlinear transformation. The entire FFN is a nonlinear transformation of the model, which complements the global interaction mechanism of Attention to jointly build a powerful context-aware representation. Subsequently followed by Add & LN to supplement the feature information and accelerate the model convergence, each subsequent layer of Transformer Encoder Layer is a stacking effect to deepen the Attention weight of the feature vectors, and the output of the whole Holistic Encoder is $\{f^{\mathcal{H}}, \boldsymbol{x}^{\mathcal{H}}\} = \mathcal{H}(F^{\mathcal{F}}), f^H \in \mathbb{R}^D, x^{\mathcal{H}} \in R^{N \times D}$.

3.5 Conv-Module

The image features are subjected to feature learning by Holistic Encoder capturing the relative position information of the elements in the sequence, but it has the limitation of learning absolute position information. The signature verification task needs to be accurate to the stroke position information (e.g., stroke order, local structural differences), so after this in order to enhance some small portion of the image features, the output $\boldsymbol{x}^{\mathcal{H}}$ of the patch token of the Holistic Encoder is rearranged, and the flattened token is reconverted into a 2D shape $\mathbf{F}^{\mathcal{H}} \in \mathbb{R}^{h \times w \times D}$

to perform a convolution operation, where $h = \lfloor \frac{H/4-P+S}{S} \rfloor, w = \lfloor \frac{W/4-P+S}{S} \rfloor$. This module introduces the convolution operation to reprocess the patch features output from the Transformer encoder to enhance the position-awareness. In the Conv-Module architecture, the Conv design of TransOSV [40] is referenced, and the number of output channels of the intermediate convolutional 2D layer is adjusted to increase the downsampling operation according to the style of previous CNNs, as shown in Fig. 3.7.

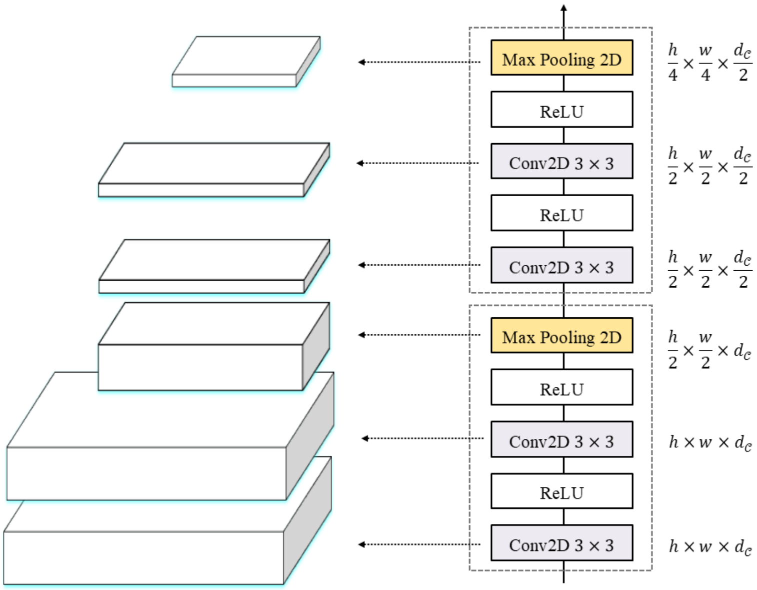

Max Pooling 2D
ReLU
Conv2D 3 × 3
ReLU
Conv2D 3 × 3
Max Pooling 2D
ReLU
Conv2D 3 × 3
ReLU
Conv2D 3 × 3
$\frac{h}{4} \times \frac{w}{4} \times \frac{d_c}{2}$
$\frac{h}{2} \times \frac{w}{2} \times \frac{d_c}{2}$
$\frac{h}{2} \times \frac{w}{2} \times \frac{d_c}{2}$
$\frac{h}{2} \times \frac{w}{2} \times d_c$
$h \times w \times d_c$
$h \times w \times d_c$

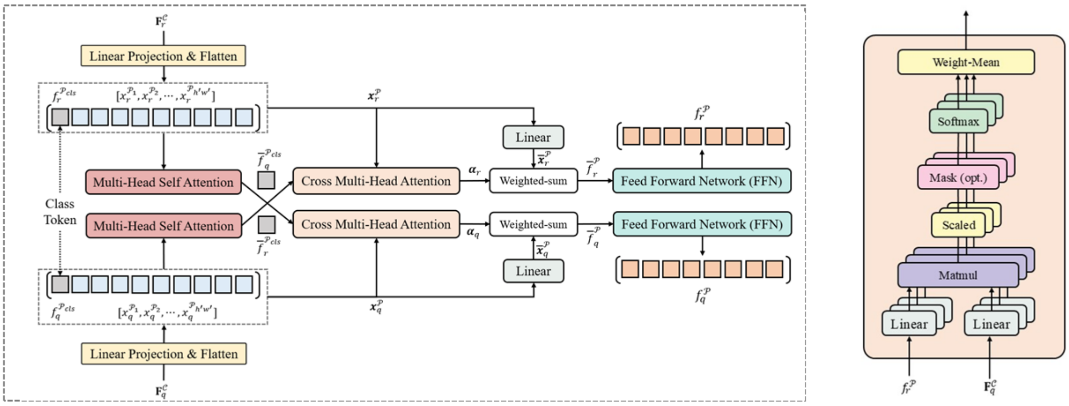
Fig. 3.7: Conv-Module structure

Different from the traditional CNN with gradually more channels, the overall downsampling operation thinking is adopted here, and certain downsampling operations are performed on the patch features to abstractly express the local features of the feature map on the basis of integrating the local detail features, with the purpose of enhancing the generalization ability and robustness of the model. This operation is added on the basis of backbone and FPN Fusion to further strengthen the model's ability to perceive the local features of the image, strengthen the focus on the key local regions of the signature, and filter the background interference at the same time. In the Conv-Module module, each Conv2D layer operation is the same as the Conv2D layer operation in backbone, which takes the combination of Conv2D+ReLU, and after two Conv2D+ReLUs a downsampling operation, i.e., Max Pooling 2D operation, will be performed to integrate the feature maps and eliminate the non-important parts of the features to integrate and eliminate non-important parts of the feature map and enhance feature expressiveness.

In the model training process, the output $\mathbf{F}^{\mathcal{C}} = \mathcal{C}(\mathbf{F}^{\mathcal{H}}) \in \mathbb{R}^{h' \times w' \times d'_{\mathcal{C}}}$ of the Conv-Module will undergo a Global Average Pooling (GAP) operation while entering the Contrast based Part Decoder to globally sample the feature maps of the output of the Conv-Module. The feature maps of the Conv-Module are globally sampled to obtain a flat local convolutional feature $f^{\mathcal{C}}$ to complete the corresponding feature extraction in the training and inference process.

3.6 Contrast based Part Decoder

This part is an improvement on TransOSV's Contrast based Part Decoder [40] for parallel computation of Cross-Attention, which references the multi-head attention computation mechanism of MHSA in Holistic Encoder. The decoder is shown in Fig. 3.8.


$\mathbf{F}_r^{\mathcal{C}}$
Linear Projection & Flatten
$f_r^{\mathcal{P}_{cls}}$
$[x_r^{\mathcal{P}_1}, x_r^{\mathcal{P}_2}, \cdots, x_r^{\mathcal{P}_{h'w'}}]$
$\boldsymbol{x}_r^{\mathcal{P}}$
Linear
$\overline{\boldsymbol{x}}_r^{\mathcal{P}}$
$f_r^{\mathcal{P}}$
Class Token
Multi-Head Self Attention
$\overline{f}_q^{\mathcal{P}_{cls}}$
Cross Multi-Head Attention
$\boldsymbol{\alpha}_r$
Weighted-sum
$\overline{f}_r^{\mathcal{P}}$
Feed Forward Network (FFN)
Multi-Head Self Attention
$\overline{f}_r^{\mathcal{P}_{cls}}$
Cross Multi-Head Attention
$\boldsymbol{\alpha}_q$
Weighted-sum
$\overline{f}_q^{\mathcal{P}}$
Feed Forward Network (FFN)
Linear
$\overline{\boldsymbol{x}}_q^{\mathcal{P}}$
$f_q^{\mathcal{P}}$
$f_q^{\mathcal{P}_{cls}}$
$[x_q^{\mathcal{P}_1}, x_q^{\mathcal{P}_2}, \cdots, x_q^{\mathcal{P}_{h'w'}}]$
$\boldsymbol{x}_q^{\mathcal{P}}$
Linear Projection & Flatten
$\mathbf{F}_q^{\mathcal{C}}$
Weight-Mean
Softmax
Mask (opt.)
Scaled
Matmul
Linear
Linear
$f_r^{\mathcal{P}}$
$\mathbf{F}_q^{\mathcal{C}}$

Contrast based Part Decoder

Cross Multi-Head Attention

Fig. 3.8: Contrast based Part Decoder structure

The decoder is similar to the Holistic Encoder part, the feature maps of references and query are mapped to the $d_{\mathcal{P}}$ feature dimension in the input phase, followed by flattening and addition of a learnable weight $f_r^{\mathcal{P}_{cls}}, f_q^{\mathcal{P}_{cls}} \in \mathbb{R}^{d_{\mathcal{P}}}$, also defined as a class token, i.e., the input part is obtained as a pair of flattened feature map vectors $\boldsymbol{x}_r^{\mathcal{P}} = [f_r^{\mathcal{P}_{cls}}, x_r^{\mathcal{P}_1}, \cdots, x_r^{\mathcal{P}_{h'w'}}]$, $\boldsymbol{x}_q^{\mathcal{P}} = [f_q^{\mathcal{P}_{cls}}, x_q^{\mathcal{P}_1}, \cdots, x_q^{\mathcal{P}_{h'w'}}] \in \mathbb{R}^{h' \times w' \times (d_{\mathcal{P}}+1)}$. Subsequently, MHSA operations are performed on $\boldsymbol{x}_r^{\mathcal{P}}, \boldsymbol{x}_q^{\mathcal{P}}$ to take out the class token $\overline{f}_r^{\mathcal{P}_{cls}}, \overline{f}_q^{\mathcal{P}_{cls}} \in \mathbb{R}^{d_{\mathcal{P}}}$ of their respective output eigenvectors as the query matrix cross-input Cross Multi-Head Attention (CMHA).The structure of CMHA is shown in Figure 8-b shown, differs from MHA in that only the query matrix and key matrix are generated to compute their cross-attention weights $\boldsymbol{\alpha}_r, \boldsymbol{\alpha}_q$, and the final concat is replaced with a weighted average to handle the attention weights of multiple heads. For a single CMHA the output cross-attention weight $\boldsymbol{\alpha}$ is calculated as eq. 3.11.

$$\boldsymbol{\alpha} = \frac{1}{M} \sum_{m=1}^{M} \text{Softmax} \left(\frac{\boldsymbol{X}_{\mathrm{q}} \mathbf{T}^{\mathcal{P}}_{m,1} \cdot (\boldsymbol{X}_{\mathrm{k}} \mathbf{T}^{\mathcal{P}}_{m,2})^{\mathrm{T}}}{\sqrt{d'_{\mathcal{P}}}} \right) \tag{3.11}$$

where $\boldsymbol{X}_{\mathrm{q}}, \boldsymbol{X}_k$ denote the input query and key vectors, $\mathbf{T}^{\mathcal{P}} \in \mathbb{R}^{M \times 2 \times D \times d'_{\mathcal{P}}}$ denotes the linear mapping layer weights of the input part of the CMHA. The resulting $\boldsymbol{\alpha}_r, \boldsymbol{\alpha}_q$ computation process is as eq. 3.12.

$$\begin{aligned} \boldsymbol{\alpha}_r &= \frac{1}{M} \sum_{m=1}^{M} \text{Softmax} \left(\frac{\overline{f}_q^{\mathcal{P}_{cls}} \mathbf{T}^{\mathcal{P}}_{m,1} \cdot (\boldsymbol{x}^{\mathcal{P}}_r \mathbf{T}^{\mathcal{P}}_{m,2})^{\mathrm{T}}}{\sqrt{d'_{\mathcal{P}}}} \right) \\ \boldsymbol{\alpha}_q &= \frac{1}{M} \sum_{m=1}^{M} \text{Softmax} \left(\frac{\overline{f}_r^{\mathcal{P}_{cls}} \mathbf{T}^{\mathcal{P}}_{m,1} \cdot (\boldsymbol{x}^{\mathcal{P}}_q \mathbf{T}^{\mathcal{P}}_{m,2})^{\mathrm{T}}}{\sqrt{d'_{\mathcal{P}}}} \right) \end{aligned} \tag{3.12}$$

Perform a weight-sum operation on the attention weights and the mapped flat feature vector $\overline{\boldsymbol{x}}^{\mathcal{P}}_r, \overline{\boldsymbol{x}}^{\mathcal{P}}_r \in \mathbb{R}^{h' \times w' \times d_{\mathcal{P}}}$, and fuse to generate the flat feature $\overline{f}_r^{P}, \overline{f}_q^{P} \in \mathbb{R}^{d_P}$ with cross-attention weights. Its after FFN to get the flat feature $\overline{f}_r^{\mathcal{P}}, \overline{f}_q^{\mathcal{P}} \in \mathbb{R}^{d_P}$ of cross attention of decoder, the whole computation process is as eq. 3.13.

$$\{f^{\mathcal{P}}_r, f^{\mathcal{P}}_q, \boldsymbol{\alpha}_r, \boldsymbol{\alpha}_q\} = \mathcal{P}(F^{\mathcal{C}}_r, F^{\mathcal{C}}_q) \tag{3.13}$$

This Contrast based Part Decoder, which is similar to the structure of Transformer Decoder [3], facilitates the generation of local feature information that distinguishes between genuine and forged signatures, and emphasizes the feature relationship between a pair of data samples, enhancing the model's learning of the features between a pair of data in a sample. During the training process, the output cross-attention weights $\boldsymbol{\alpha}_r, \boldsymbol{\alpha}_q$ are computed with sparsity loss to force the distribution of cross-attention weights to be centralized, avoiding averaging in order to fail to pay attention to the sensitive part of forged signatures.

3.7 Loss Funcion

3.7.1 Euclidean Distance

OSVTF is a model architecture similar to twin networks, this style emphasizes that the model is a two-flow form of shared weights and the overall purpose is to compare the gap between the two, so it will be based on the similarity of the input two features in order to design the loss function. In this work, Euclidean Distance [27] will be taken in order to be used as a similarity calculation for a pair of feature samples, assuming that a pair of features f_r, f_q is input and the formula for calculating the distance between them is as eq. 3.14.

$$\mathcal{D}(f_r, f_q) = ||f_r - f_q||_2 = \sqrt{\sum_{d=1}(f_r^{(d)} - f_q^{(d)})^2} \tag{3.14}$$

3.7.2 Focal Contrast Loss

In the handwritten signature sample data processing stage, the handwritten signature of the writer will be paired with the pairing operation that will not be repeated between the two, for example, the BHSig-B [4] dataset contains a total of 100 writers' handwritten signature images, in which each writer owns 24 genuine signature images ($y = 0$) and 30 forged signature images ($y = 1$), the genuine signature images are used as the reference signature, forged signature and real signature together as query signature, which will generate 276 pairs of positive samples and 720 pairs of negative samples. Thus each writer has a total of 996 data sample pairs, in which the ratio between positive and negative sample pairs is not close to 50:50, so the number of positive and negative samples is not balanced and belongs to the category of hard samples [2].

The loss function of similarity of comparison samples used for binary classification in earlier algorithms and models is Contrast Loss [28], which is calculated as eq. 3.15.

$$\mathcal{L}_c(f_r, f_q) = (1 - y) \cdot \mathcal{D}(f_r, f_q)^2 + y \cdot \text{max } (m - \mathcal{D}(f_r, f_q), 0)^2 \tag{3.15}$$

Where m denotes the boundary value, i.e., the distance that forged samples should be kept at least. For the sample pair of $y = 0$, it is directly used as the loss value to make the model parameter learning on the real sample pairs of features more similar; on the contrary $y = 1$, if the feature distance between the reference and query is less than m, $(m - \mathcal{D}(f_r, f_q))^2$ is used as the loss value to make the model for the forged signature pairs of features is not less than m. However, this kind of loss can't be dynamically emphasized on the hard samples and is prone to model parameter overfitting problems. In order to reduce the risk of overfitting, a boundary value is also introduced for the genuine sample pairs to reduce the unnecessary contraction between the genuine pairs, i.e., the Double-Margin loss [37] is calculated as eq. 3.16.

$$\begin{aligned}\mathcal{L}_{dm}(f_r, f_q) = (1-y) \cdot \max(\mathcal{D}(f_r, f_q) - n, 0)^2 \\ + y \cdot \max\ (m - \mathcal{D}(f_r, f_q), 0)^2\end{aligned} \tag{3.16}$$

where the two boundary values m>n, penalize the case where the distance exceeds n for the genuine sample pairs, and promote the case where the distance is greater than m for the forged sample pairs. Although double margin optimizes the problem of easy overfitting by contrast loss to a certain extent, in the case of two pairs of sample data $\{r, q_1\}, \{r, q_2\}$ and $y = 1$ In the case, the distance between the two sample pairs is calculated by OSVTF, if $\mathcal{D}(f_r, f_{q_1}) >> \mathcal{D}(f_r, f_{q_2}) > n$, then the loss function should give a greater loss to r, q_1, but $\mathcal{L}_{dm}$ will treat the two sample pairs equally, and the notion of dynamic weighting is introduced in TransOSV, named Focal Contrast Loss [40], which is calculated as eq. 3.17.

$$\begin{aligned}\mathcal{L}_{fc}(f_r, f_q) = (1-y) \cdot \sigma(\overline{K}(\mathcal{D}(f_r, f_q) - \alpha_1)) \cdot \max\ (\mathcal{D}(f_r, f_q) - n, 0)^2 \\ + y \cdot \sigma(\overline{V}(\alpha_2 - \mathcal{D}(f_r, f_q)) \cdot \max\ (m - \mathcal{D}(f_r, f_q))^2\end{aligned} \tag{3.17}$$

where $\sigma(\boldsymbol{z}) = \frac{1}{1+e^{-\boldsymbol{z}}}$, denotes the Sigmoid activation function [33], which is used to dynamically generate the weights; α_1, α_2 are the two margin values; $\overline{K}, \overline{V}$ denote the scaling factors, which are used to regulate the scaling factor of the response strength of hard samples. When the label of the sample pair is genuine, if the feature distance is greater than α_1, the current weight value is larger; when the label is forged, if the feature distance is smaller than α_2, the weight is larger. As a result, $\mathcal{L}_{fc}$ will have the advantages of dynamically adjusting the attention of the training samples, focusing on optimizing the “easily confused” signatures, and improving the generalization ability of the model, and it will be used as one of the main training loss functions of the OSVTF model. According to the flat features extracted from each module, four parts will be taken to calculate the focal contrast loss, and the overall Focal Contrast Loss is calculated as eq. 3.18.

$$\mathcal{L}_{fc}(f_r, f_q) = \begin{bmatrix}\lambda_1, \lambda_2, \lambda_3, \lambda_4\end{bmatrix} \cdot \begin{bmatrix}\mathcal{L}_{fc}(f_r^{\mathcal{F}}, f_q^{\mathcal{F}}) \\ \mathcal{L}_{fc}(f_r^{\mathcal{H}}, f_q^{\mathcal{H}}) \\ \mathcal{L}_{fc}(f_r^{\mathcal{C}}, f_q^{\mathcal{C}}) \\ \mathcal{L}_{fc}(f_r^{\mathcal{P}}, f_q^{\mathcal{P}})\end{bmatrix} \tag{3.18}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are training hyperparameters used to assign the weight of each feature.

3.7.3 Sparsity Loss

In Contrast based Part Decoder, for the cross attention calculation, the attention weights will be too uniformly distributed, in fact, the model should be enhanced to focus on the most significant local differences in the region, and thus the cross-entropy approach is adopted to generate the entropy constraints of the contrast mask in the hope that the cross-attention weights can be sparsely distributed, thus enhancing the Decoder’s local attention accuracy [40], which is calculated as eq. 3.19.

$$\mathcal{L}_{spa}(\boldsymbol{\alpha}_r, \boldsymbol{\alpha}_q) = -\sum_{i=1}^{h'w'} \boldsymbol{\alpha}_q^{(i)} \log\ (\boldsymbol{\alpha}_q^{(i)}) - \sum_{i=1}^{h'w'} \boldsymbol{\alpha}_r^{(i)} \log\ (\boldsymbol{\alpha}_r^{(i)}) \tag{3.19}$$

Since this component is not trained as the main model, ξ is introduced as the hyperparameter of this component to participate in OSVTF training. In summary, the overall loss function of OSVTF is composed as eq. 3.20.

$$\mathcal{L} = \mathcal{L}_{fc}(f_r, f_q) + \xi \mathcal{L}_{spa}(\boldsymbol{\alpha}_r, \boldsymbol{\alpha}_q) \tag{3.20}$$

Chapter 4 Experiments

4.1 Experimental Setup

In this work, we will build OSVTF based on ViT [5] configuration.The output feature map of The FPN Fusion has the same number of feature dimensions as the number of channels in the first layer of the scaled feature convolution map.The encoder contains 8 Transformer encoder layers ($L \times 8$) and each MHSA in the encoder layer contains 12 heads ($M = 12$), 768 features ($D = 768$), and a sliding window of size 4×4 steps of 2 ($P = 4, S = 2$). In the Conv-Module part, set the number of channels in its intermediate convolutional layer to 512 ($d_{\mathcal{C}} = 512$). In the decoder part, the number of feature dimensions of MHSA and CMHA, and the number of headers are consistent with the parts of encoder, and the number of feature dimensions of weighted-sum's mapping of patch embeddings after CMHA is set to 512 ($d_P = 512$).

On the loss function, the partial settings $n = 0.3, m = 0.9, \alpha_1 = 0.6, \alpha_2 = 0.8$ and $\overline{K} = \overline{V} = 10$ are followed for TransOSV.The hyper-parameters are set to $\lambda_1 = 0.5, \lambda_2 = 1.0, \lambda_3 = 0.5, \lambda_4 = 2.0, \xi = 0.2$.On the optimizer, all experiments are On the optimizer, all experiments are conducted on 2 NVIDIA 3080 GPUs and the total batch size is set to be 16.

4.2 Metrics

In terms of evaluation metrics, unlike conventional binary classification tasks, the offline handwritten signature verification task requires special attention to negative samples, and therefore standard metrics in biometric and signature verification tasks [21] are used to evaluate the performance of OSVTF, i.e., False Rejection Rate (FRR), False Acceptance Rate (FAR) and Equal Error Rate (EER). In a conventional binary classification task, the concept of confusion matrix [34] is used to count the number of sample labels and predicted labels in the format of Table 4.1.

Table 4.1: Confusion Matrix

predicted Label / raw label	$\hat{y} = 0$ (positive)	$\hat{y} = 1$ (negative)
$y = 0$	True Positive (TP)	False Negative (FN)
$y = 1$	False Positive (FP)	True Negative (TN)

This leads to the following formula for FRR, and FAR as eq. 4.1:

$$\begin{aligned} \mathrm{FRR} &= \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} \\ \mathrm{FAR} &= \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \end{aligned} \tag{4.1}$$

The EER is calculated using several thresholds and distances for comparison, if the feature distance of the sample pair is greater than the threshold, then its category is forged, otherwise it is genuine. This work will calculate the distance based on the model features of the sample pairs of the batch data during the training process, and based on the distance minima and maxima of the sample pairs in order to set a number of thresholds, so that the FAR and FRR are automatically calculated for the current batch for each threshold value. FAR and FRR are calculated for each threshold and when the difference between FAR and FRR is the smallest, then FRR, FAR for the current threshold are output simultaneously and EER is calculated as eq. 4.2:

$$\mathrm{EER} = \frac{\mathrm{FRR} + \mathrm{FAR}}{2} \tag{4.2}$$

As a result, the experimental phase will be based on the ACC, FRR, FAR and EER to comprehensively assess the model performance.

4.3 Dataset

BHSig-B and BHSig-H. Both datasets are published by Indian Institute of Technology Guwahati (IIT Guwahati) [4] dataset of handwritten signatures in Bengali and Hindi. Where BHSig-B contains Bengali handwritten signatures of 100 writers and BHSig-H contains Hindi handwritten signature images of 160 writers. Each user handwritten signature has signed 24 authentic and 30 forged signatures.

CEDAR. This dataset is an English handwritten signature dataset developed and published by Center of Excellence for Document Analysis and Recognition [30]. It contains 55 handwritten signatures of writers in English. Each user signed 24 authentic and 24 forged signatures by hand.

4.4 Results Analysis

4.4.1 Writer-Independent Results

On the WI task, OSVTF takes the Euclidean distance method to calculate the distance between two features to determine whether the signature is forged or not, in order to evaluate whether OSVTF outperforms the past models in possessing offline handwritten signature verification performance. The initial stage is validated on BHSig-B and BHSig-H, which are consistent with the past algorithms, by dividing the 100 writers of BHSig-B into training and testing sets according to the ratios of 50:50 and 80:20 for 100 writers, and dividing the 160 writers of BHSig-H into training and testing sets according to the ratio of 100:60 for 160 writers. The preliminary experimental results obtained are shown in Table 4.2.

Table 4.2: BHSig dataset WI task performance comparison

Model	FRR	FAR	EER
BHSig-B 50/50			
SigNet [30]	13.89	13.89	13.89
CaP [37]	3.96	3.96	3.96
IDN [23]	5.24	4.12	-
InceptionSVGNet [29]	**2.22**	**3.88**	-
TransOSV [40]	9.90 (12.41)	9.90 (12.41)	9.90 (12.41)
OSVTF [Ours]	10.72	10.85	10.79
BHSig-B 80/20			
DeepHsv [6]	11.92	11.92	11.92
TransOSV [40]	**3.56** (5.11)	**3.56** (5.11)	**3.56** (5.11)
OSVTF [Ours]	4.47	4.61	4.54
BHSig-H 100/60			
SigNet [30]	15.36	15.36	15.36
IDN [23]	4.93	8.99	-
CaP [37]	5.97	5.97	5.97
InceptionSVGNet [29]	3.33	6.38	-
TransOSV [40]	**3.24** (4.86)	**3.24** (4.86)	**3.24** (4.86)
OSVTF [Ours]	3.86	3.94	3.90

Among them, in the BHSig-B dataset, 10.79% EER for OSVTF at 50:50 ratio and 12.41% EER for reproduced TransOSV, with a difference of about 15.01%, and 4.54% EER for OSVTF at 80:20 ratio and 5.11% EER for reproduced TransOSV, with a difference of about 12.56%. In the BHSig-H dataset, 4.86% EER for reproduced TransOSV, 3.90% EER for OSVTF, a difference of about 24.62%. The OSVTF performance on the WI task outperforms the TransOSV performance by about 17%. The performance of OSVTF is slightly inferior to that of TransOSV in the above experimental environments, but the performance of OSVTF compared to TransOSV in the same environments has some improvement, confirming that the multi-scale fusion features of an image have a certain degree of enhancement in the learning ability of the image features on the basis of the combined features. In the BHSig-B 50/50 dataset, OSVTF obtains far less results than CaP, IDN, and InceptionSVGNet, but outperforms the same type of models on the BHSig-B 80/20 and BHSig-H 100/60 datasets, to a certain extent, because these networks are the specialized networks for that scale dataset, and therefore can reflect OSVTF in cross-dataset scenarios is able to adapt to the variations in the images of each dataset to a large extent.

4.4.2 Writer-Dependent Results

On the WD task, in addition to the experiments on the model with the addition of multiscale fusion features, a classifier for the deep learning traditional image classification task (GAP classifier) was added to verify if it is a better fit compared to the SVM's classifier for the model with multiscale fusion features. Therefore, the 80:20 ratio of BHSig-B and BhSig-H were validated and the results are shown in Table 4.3.

Table 4.3: BHSig-B dataset WD task performance comparison

Model	FRR	FAR	EER
BHSig-B 80/20			
TransOSV [Ours]	3.78	3.78	3.78
OSVTF [SVM]	2.74	2.80	2.77
OVSTF [GAP]	**2.05**	**2.21**	**2.13**
BHSig-H 100/60			
TransOSV [Ours]	4.23	4.23	4.23
OSVTF [SVM]	3.17	3.21	3.19
OVSTF [GAP]	**2.67**	**2.69**	**2.68**

Using SVM as a classifier, in the experimental results on the BHSig-B dataset, OSVTF 2.77% EER decreased by 36.46% compared to TransOSV's 3.78% EER, and in the experimental results on the BHSig-H dataset, OSVTF 3.17% EER decreased compared to TransOSV 4.23% EER by 33.44%. The OSVTF performance on the WD task outperforms the TransOSV performance by about 34.95%. On the experimental results of different classifiers, OSVTF 2.13% EER of BHSig-B taking GAP classifier is better than OSVTF 2.77% EER taking SVM classifier; OSVTF 2.68% EER of BHSig-H taking GAP classifier is better than OSVTF 3.19% EER taking SVM classifier. on the overall WD task, OSVTF of OSVTF also taking SVM classifier outperforms TransOSV by about 34.95%, and can improve by another 19.03% when taking GAP classifier. From this, it can be judged that for OSVTF that have adopted multi-scale fusion features, the author classifier that adopts GAP can better handle the above features compared to SVM.

4.5 Ablation Experiments

In the above experimental results, it is shown that OSVTF with the addition of multi-scale fusion features improves the performance on the WI task by about 17% and on the WD task by about 35% (and by about 45% when GAP classification is taken) compared to TransOSV. In this part of the work, ablation experiments will be conducted for the multiscale fusion features, Conv-Module downsampling method, decoder multicast optimization, and sparsity loss in order to verify the performance enhancement of each part for OSVTF, and the obtained

WI experimental results are shown in Table 4.4.

Table 4.4: WI task EER results of ablation experiments

Model	$\mathcal{F}$	$\mathcal{C}$	$\mathcal{P}$	$\mathcal{L}_{spa}$	BHSig-B(80/20)	BHSig-H(100:60)	CEDAR(50:5)
TransOSV					5.11	4.86	5.94
Baseline	√				4.71	4.39	5.40
		√			4.83	4.55	5.67
			√		4.92	4.76	5.61
	√	√	√		4.64	4.21	5.21
OSVTF	√	√	√	√	**4.54**	**3.90**	**4.75**

It can be shown that the addition of fusion features of multi-scale feature maps to TransOSV can substantially reduce the model misclassification rate. The Conv-Module with downsampling approach (with simultaneous addition of input mapping in the decoder) enhances the abstract representation of features for more complex modeling operations. In the multi-head optimization of the decoder, multiple perspectives are added to focus on the feature vectors, so the model performance is improved. In the sparsity loss training scheme, the distribution of attention in the decoder is slightly weakened, allowing more attention to be paid to individual local features of the signature image.

In the Conv-Module and decoder part of this work, the mapping structure of dimensionality reduction $\rightarrow$ dimensionality enhancement is designed to change the whole Conv-Module from the original change in the number of feature dimensions$D \rightarrow d_{\mathcal{C}} \rightarrow D$ to $D \times d_{\mathcal{C}} \times \frac{d_{\mathcal{C}}}{2}$. In order to comply with this change a ”dimensionality enhancement” linear mapping layer is introduced at the Encoder input stage. In order to comply with this change, a ”dimension-up” linear mapping layer is introduced at the Encoder input stage. In order to verify the feasibility of the ”dimensionality reduction→dimensionality enhancement” mapping structure, a large learning rate and 20 training cycles are set in the experiments on the BHSig-B 50/50 dataset, and the changes in the loss function and evaluation indexes before optimization are shown in Fig. 9, and the changes in the loss function and evaluation indexes after optimization are shown in Figs. 4.1 and 4.2.

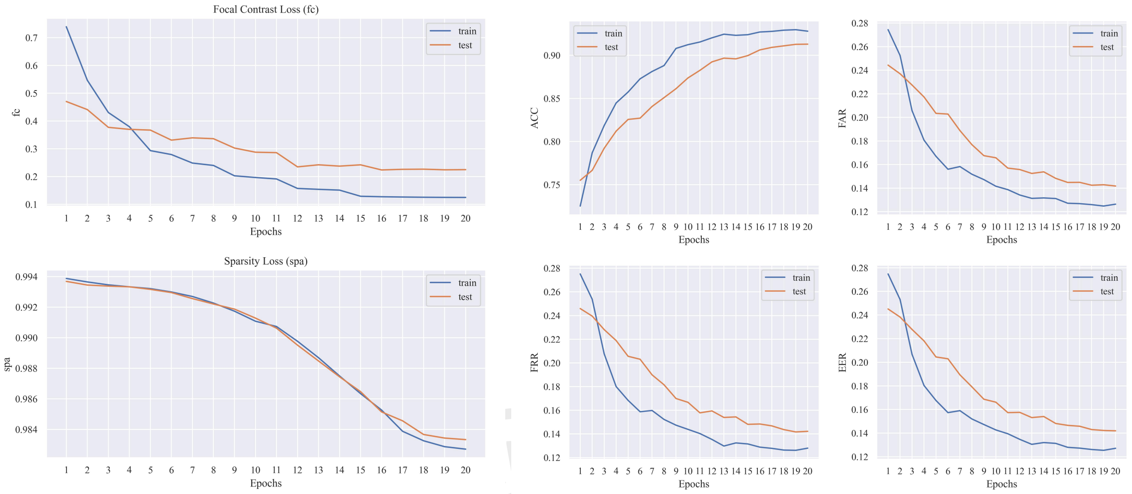


Fig. 4.1: Previous Conv-Module loss and metrics diversification

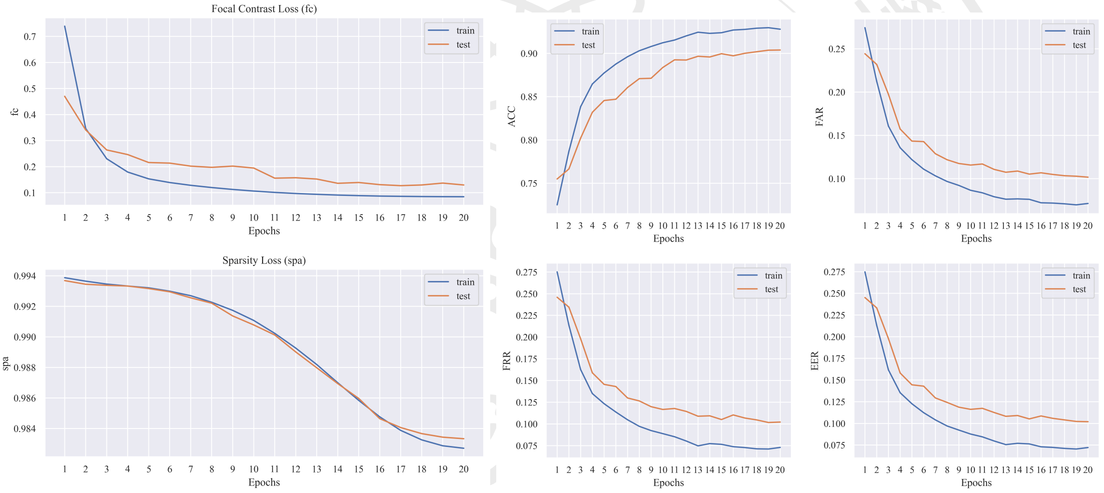


Fig. 4.2: Optimized Conv-Module loss and metrics diversification

Based on the results, the parameters of the pre-optimization OSVTF start converging at about the 7-8th Epoch in the proposed experimental scheme, while the parameters of the optimized OSVTF start converging at about the 3-4th Epoch. This feature processing method of downsampling followed by mapping can effectively accelerate the convergence speed of the model training process. Since the number of features extracted from the intermediate network layer of OSVTF is large, this approach can learn the sensitive part of forged signature features more quickly, and combined with the computation of CMHA, it further strengthens the learning of the features of the sensitive part of the forged signature, and this approach can reduce a certain number of model parameters, which makes the size of the model weight file become smaller, and the inference speed is accelerated.

Chapter 5 Conclusions

In this work, a deep learning model architecture for image multi-scale feature fusion based on CNN+Transformer style is proposed for the current stage of deep learning models for offline handwritten signature verification. The model architecture is further improved on the basis of the TransOSV model architecture similar to the twin network style by adding the feature of extracting different scale feature maps from multiple network layers in the CNN on the basis of multiple different feature combination mechanisms adopted, and using the FPN Fusion module of FPN-Style to fuse the multi-scale features. Secondly, the Conv-Module and Part Decoder of the sub-network were optimized to adapt the multi-scale fusion features. In order to accelerate the speed of model inference with the added multiscale fusion features, a downsampling-like adjustment is taken to the Conv-Module, and after reconstructing the feature vectors of the Holistic output into 2D image features, a convolution operation with downsampling is taken in the feature dimensions to optimize the redundancy of the useless features in the image by decreasing the number of channels and the size of the image through this part. In Part Decoder, a mapping operation is added to flatten the image features from the Conv-Module downsampling operation and map them into the higher dimensions with a modified cross-multiple attention computation, thus accelerating the overall inference process in a model architecture with the addition of a feature extraction sub-network.

In the experimental phase part, it is total divided into three parts: replication experiments on the WI task, comparison experiments of classifiers on the WD task, and model ablation experiments. On the WI task, the BHSig-B and BHSig-H datasets are taken to train the initially constructed OSVTF with certain epochs. For the BHSig-B dataset, the 50/50 ratio is to evaluate the generalization ability of the model, and the 80/20 ratio is to evaluate the comprehensive learning ability of the model. On the WD task, the focus is on comparing the difference between the GAP classifier and the SVM classifier. In the experimental part of the model, a side-by-side comparison is made for the optimization of multi-scale fusion features, Conv-

Module, and part decoder. The experimental results show that the optimization of Conv-Module and part decoder after adopting multi-scale fusion features can make the parameters converge faster during model training.

In the side-by-side comparison, the performance of OSVTF has outperformed the previous TransOSV model. However, in the model training process, the performance of TransOSV model cannot be perfectly reproduced due to the limitation of the training environment and the lack of model details. Meanwhile, in the offline signature verification task, there is no special preprocessing of the image, the image size is set to be slightly larger than that of ImageNet, and a weakened ResNet-34 is taken on the backbone for training. Therefore, in the next step, we will try to set up more GPUs and set up fixed image sizes and image preprocessing work for different ResNet series models on the backbone, so as to experiment with different specifications of the OSVTF model, which can work in various device environments as much as possible.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *in Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*, Curran Associates Inc., 2012, pp. 1097–1105.

[2] A. Shrivastava, A. Gupta, and R. Girshick, “Training Region-based Object Detectors with Online Hard Example Mining,” *in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 761–769.

[3] A. Vaswani et al., “Attention is All You Need,” *in Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[4] A. Bhatawdekar, S. Bhattacharya, R. Khatri, and R. Tiwari, “BHSig260: A Dataset for Offline Signature Verification,” 2020.

[5] A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *in Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[6] C. Li, F. Lin, Z. Wang, C. Yu, L. Yuan, and H. Wang, “Deephsv: User-independent offline signature verification using two-channel CNN,” *in International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

[7] D. Hendrycks and K. Gimpel, “Gaussian Error Linear Units (GELUs),” 2016.

[8] D. Banerjee, K. Dasgupta, D. Ganguly, and K. Chatterjee, “A Survey of Offline Handwriting Signature Recognition,” Mar. 2019.

[9] E. J. R. Justino, F. Bortolozzi, and R. Sabourin, “A comparison of SVM and HMM classifiers in the off-line signature verification,” *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1377–1385, 2005.

[10] H. Li S. Zhang and M. Ouyang, “Multimodal Signature Verification via Hybrid CNN-Transformer Networks,” *in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 11230–11239.

[11] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning. MIT Press,” 2016.

[12] J. F. Vargas, M. A. Ferrer, C. M. Travieso, and J. B. Alonso, “Off-line Handwritten Signature GPDS-960 Corpus,” *in Proc. 9th International Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 764–768.

[13] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization,” 2016.

[14] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *in Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[17] L. L. Lee, T. Berger, and E. Aviczer, “Reliable on-line human signature verification systems,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 643–647, Jun. 1996.

[18] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, “Learning features for offline handwritten signature verification using deep convolutional neural networks,” *Pattern Recognition*, vol. 70, pp. 163–176, 2017

[19] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, “Offline handwritten signature verification —Literature review,” *in 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Nov. 2017, pp. 1–8.

[20] M. H. M. Yusof and V. K. Madasu, “Signature Verification and Forgery Detection System,” *in Proc. Student Conference on Research and Development (SCORED 2003)*, 2003, pp. 9–14.

[21] M. A. Ferrer, J. B. Alonso, and C. M. Travieso, “Offline Geometric Parameters for Automatic Signature Verification Using Fixed-Point Arithmetic,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 993–997, 2005.

[22] M. Lin, Q. Chen, and S. Yan, “Network In Network,” *in Proc. International Conference on Learning Representations (ICLR)*, 2014.

[23] M. Wang, S. Lu, D. Zhu, and J. Lin, “A High-Speed and Low-Complexity Architecture for Softmax Function in Deep Learning,” *in 2018 IEEE*

Asia Pacific Conference on Circuits and Systems (APCCAS), Chengdu, China: IEEE, 2018, pp. 223–226.

[24] P. Wei, H. Li, and P. Hu, “Inverse discriminative networks for handwritten signature verification,” *in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] R. G. Keys, “Cubic Convolution Interpolation for Digital Image Processing,” vol. 29, 1981.

[26] R. Plamondon and G. Lorette, “Automatic signature verification and writer identification—The state of the art,” *Pattern Recognition*, vol. 22, no. 2, pp. 107–131, 1989.

[27] R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern Classification,” *2nd ed. Wiley-Interscience*, 2000.

[28] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality Reduction by Learning an Invariant Mapping,” *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 1735–1742.

[29] R. K. Mohapatra, K. Shaswat, and S. Kedia, “Offline handwritten signature verification using CNN inspired by inception V1 architecture,” *in 2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019, pp. 263–267.

[30] S. Pankanti, S. Prabhakar, and A. K. Jain, “Cedar: A database of handwritten signatures for benchmarking signature verification systems,” *in Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2000, pp. 5–8.

[31] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *in Proc. 32nd International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[32] S. Dey, A. Dutta and J. I. Toledo, “Signet: A Deep Learning Architecture for Offline Signature Verification Using Multi-Scale CNN and SVM,” *IEEE Access*, vol. 8, pp. 118942–118956, 2020.

[33] T. M. Mitchell, “Machine Learning.” *McGraw-Hill*, 1997.

[34] T. Fawcett, “An Introduction to ROC Analysis,” vol. 27. 2006.

[35] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017.

[36] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” *Proc. 27th International Conference on Machine Learning (ICML)*, pp. 807–814, 2010.

[37] X. Lu, L. Huang, and F. Yin, “Cut and Compare: End-to-End Offline Signature Verification Network,” *in Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 176–183.

[38] X. Wang, Y. Chen, Z. Liu and L. Jin, “Vision Transformer for Offline Handwritten Signature Verification,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 3, pp. 450–465, Jul. 2022.

[39] Y. Muhtar, W. Kang, A. Rexit, Mahpirat, and K. Ubul, “A Survey of Offline Handwritten Signature Verification Based on Deep Learning,” *in 2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML)*, Jul. 2022, pp. 391–397.

[40] Y. Zhang et al., “TransOSV: Offline Signature Verification with Transformers,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[41] Z. Liu et al., “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” *in Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 10012–10022.

Appendix

A.1 Train Hyper-parameters

Symbols	Values
L	8
M	12
D	768
P	4
S	2
$d_{\mathcal{C}}$	512
$d_{\mathcal{P}}$	512
n	0.3
m	0.9
α_1	0.6
α_2	0.8
$\overline{K}$	10
$\overline{V}$	10
λ_1	0.5
λ_2	1.0
λ_3	0.5
λ_4	2.0
ξ	0.2

Acknowledgements

First of all, I would like to express my ddep gratitude to Professor Xin Liu, my supervisor. She gave me a lot of encouragement during the research period, so that I could overcome the difficulties in researching difficult doubts, overcome psychological laziness, and constantly challenge and surpass myself. During the period from the topic selection report to the thesis, she provided many research perspectives and innovations, gave me many key suggestions during the research period, carefully examined the thesis and made relevant suggestions for modification, which enabled me to better improve the content of each part, and not only enabled me to acquire more knowledge, but also knew how to handle various details.

Secondly, I would like to thank the professors of the School of Innovative Engineering, who taught me all they could during these two years of study, allowing me to reconstruct my previous knowledge framework from a different perspective. The lectures also gave me a stage to show my personal ability, allowing me to learn new knowledge while practicing my presentation skills, and the English-focused lectures helped me to improve my English reading and writing skills to a certain extent.

Finally, I would like to thank my family for their support and help during the study period, which helped me to calmly think about the situation and find solutions from my own point of view at every critical moment, and also made me know that I should be calm in doing things, be a good listener when listening to others, and be an eloquent speaker when talking to others.

Mingchen Wang
Macau University of Science and Technology
2025/04/29

Resume

姓名	王茗琛	入學時間	2023.09
教育背景	起止年月	就讀學校	取得學位名稱
	2019.09–2023.06	北京理工大學珠海學院	理學學士學位
在讀期間學術成果	*發表的學術論文、著作（論文/著作名稱、報刊/出版社名稱、發表時間、刊物/出版社級別）*		
	參加的學術項目 (項目名稱、項目時間、立項單位、承擔的工作)		