UTRECHT UNIVERSITY
Faculty of Science
Department of Information and Computing Sciences
MSc Artificial Intelligence

KINSHIP VERIFICATION USING VISION TRANSFORMERS

A THESIS BY
Huaixi Tang
2631695

Project supervisor Assist. Prof. dr. Itir Onal Ertugrul
Second examiner Prof. dr. Albert Salah

Abstract

Kinship verification is the term of verifying whether the given two people have a kin relationship from their facial images or videos or other biological features. As a soft bio-metric modality, visual kinship verification has high availability and extremely low cost compared to DNA-based methods. It is a huge challenge to analyze kinship based on visual information, mainly because the kin relationship has s large intra-class differences and small inter-class differences due to factors such as gender and age. This requires us to extract more discriminative features. Video data can bring us a new dimension. Previous studies have shown that people with kinship not only have similar appearances but also have similar expression patterns, which suggests that we can extract dynamic features of facial videos for kinship verification. Traditional methods use handcraft features to extract dynamic features, and some new research begins to use neural networks. Our research focuses on smiling expressions, trying to extract spatio-temporal features from facial videos using a state-of-the-art video vision transformers. We created a video vision transformer based siamese network and trained it on a face video dataset. We experimentally compare the impact of using dynamic features versus purely texture features on kinship verification. We then compared the capabilities of CNNs and ViTs in extracting facial dynamic features. We tested the performance of the model by adjusting the initialization and training methods of the model. Referring to the latest research, we developed a pre-training method based on matched expression sequences to solve the challenge brings by the small size of the dataset. Our study is trained on smiling videos provided by the UvA-NEMO dataset and presents results and analytics.

Table of Contents

1. Introduction

1.1 Research motivation

Kinship verification is the term of verifying whether the given two people have a kin relationship with their facial images or videos. As a soft bio-metric modality, visual kinship verification has high availability and extremely low cost compared to DNA-based methods. It has a powerful ability to automatically analyze massive amounts of media, leading to a variety of applications such as finding missing children (Kohli et al., 2018), criminal investigations (Lu et al., 2013), and social network analysis (Lu et al., 2014).

Fang et al. (2010) first demonstrated the possibility of using computer vision methods on image pairs to automatically verify kinship. Since then, image based kinship verification has received increasing attention. Early research focus on hand-crafted features (Xia et al., 2012a; Guo and Wang, 2012) and pre-defined image descriptors (Moujahid and Dornaika, 2019; Goyal and Meenpal, 2020). More recently, deep learning methods (Wang et al., 2015; Zhang12 et al., 2015; Li et al., 2016; Dehghan et al., 2014) have emerged and shown powerful learning capability. These works usually contains a learned feature extractor with a two-stream structure sharing the weights, and applying several metric learning techniques (Lu et al., 2013; Wei et al., 2019).

Besides, identifying the kin-relationship from faces in videos is also an interesting research direction. It has some important practical use cases, such as surveillance systems. Compared to still images, facial videos contains additional spatial-temporal information that can be useful for kinship verification. Dibeklioglu et al. (2013) first model the spatial-temporal facial dynamic for video kinship verification tasks. The results indicate that people with kinship have both similar appearance and smiling expressions, which lighting a promising direction for further study.

Vision Transformers (ViTs) (Dosovitskiy et al., 2020; Touvron et al., 2021) adapts the powerful mechanism of self-attention (Vaswani et al., 2017) into the field of computer vision. ViTs have gained significant research attention, and a number of recent approaches have been proposed which build upon ViTs. Currently a lot of work uses this powerful tool for facial action unit detection, facial expression recognition and so on (Jacob and Stenger, 2021; Xue et al., 2021; Wang and Wang, 2021), to model the long-term relationship between parts of the face. Also, there has been efficient variant of ViTs for video analysis (Arnab et al., 2021).

In this thesis, we investigate the possibility of modeling spatio-temporal facial representations using state-of-the-art vision transformers to verify kinship from videos. Previous studies relied heavily on still images, and this study aimed to explore the role of facial dynamics, especially the facial action of smiling, in visual kin-recognition. Following Dibeklioglu et al. (2013); Dibeklioglu (2017), we investigate modeling spatial-temporal features through the vision transformers (Dosovitskiy et al., 2020; Touvron et al., 2021) for facial kinship verification. The proposed network is trained and validated on the UvA-NEMO Smile dataset (Dibeklioğlu et al., 2012), a standard dataset in the field of video kinship verification.

1.2 Research questions

Main question How does Vision Transformer-based siamese-network perform on video-based kinship verification task?

We further decompose the main question into the following sub-questions.

1. How does using spatial-temporal features compare to only using spatial features for video-based kinship verification?
   Existing literature supports that considering the spatial-temporal features may yield better results. To answer sub-RQ1, we designed a set of experiments to train a simple SVM classifier using different pre-trained feature extractors and observe the kinship verification performance when only considering static appearance and the spatial-temporal dynamics.
2. Can vision transformers learn better representations for kinship verification than convolutional neural networks?
   The first two sets of experiments we conduct will be used to answer sub-RQ2. These two sets of experiments compare the performance of ViT-based networks (ViT, ViViT) and convolution-based networks (ResNet, MobileNet-3D) in extracting visual features and processing spatio-temporal relationships.
3. How does the choice of loss function for contrastive learning influence the performance of the kinship verification model?
   By training vision transformers with different loss functions such as contrastive loss, triplet loss, and infoNCE, we evaluate the impact of different losses on the kinship verification performance and answer sub-RQ3.
4. Can pre-training methods enhance the performance of the kinship verification model?
   To solve the problem of limited data, we refer to the SOTA pre-training methods to try to improve the results of the model. The results of the last experiment will answer sub-RQ4.

2. Related Work

2.1 Automatic Kinship Analysis

Automatic kinship analysis is a challenging problem in the field of computer vision. Recent literature has investigated kinship analysis problems from different perspectives, see Figure 1.

Kinship verification has gained a lot of interest and has been researched as a fundamental problem. Most of the early kinship verification works were based on hand-crafted features, including enumeration and saliency features (Fang et al., 2010; Xia et al., 2012a,b),(Guo and Wang, 2012; Wang and Kambhamettu, 2014; Goyal and Meenpal, 2018) and hand-crafted image descriptors (Freitas Pereira et al., 2012; Moujahid and Dornaika, 2019; Goyal and Meenpal, 2020). Recent studies have begun to use deep learning to extract facial features (Wang et al., 2015). As a classification problem, metric learning is a common approach (Lu et al., 2013; Wei et al., 2019). Since datasets are generally small compared to mainstream tasks, many studies have used transfer learning methods to leverage data from other domains (Shao et al., 2011). At the same time, some recent studies have introduced hard example mining (Suh et al., 2019; Wang and Yan, 2020; Li et al., 2021b), which guides the network to mine discriminative information by providing more difficult-to-distinguish negative pairs to make full use of the limited positive pairs.

Many extended studies are raised based on the kinship verification. Kinship identification (Wang et al., 2020) not only determines whether the provided image pair is kin, but also specifically classifies the relationship between the two, transforming the problem into a multi-class classification problem. Common classification labels include father-son, mother-son, father-daughter, mother-daughter, and so on. The Tri-Subject kinship verification (Qin et al., 2015) uses the characteristics of children's genes inherited by both parents, provides facial images of both parents, and determines whether the child is related to these two people, which is also a binary classification problem. A lot of kinship research is limited to the information collected indoors and under the cooperation condition. Kinship verification in the wild (Robinson et al., 2018) performs kinship verification on the facial information collected without cooperation under wild conditions. This research direction has more practical applications, but at the same time challenged by changes in image quality. Additional research includes kin face synthesis (Ertugrul and Dibeklioglu, 2017), where photos of parents are provided to generate photos of children. This research has a variety of applications, including missing child matching and entertainment applications,

and can be used to enhance kinship datasets.

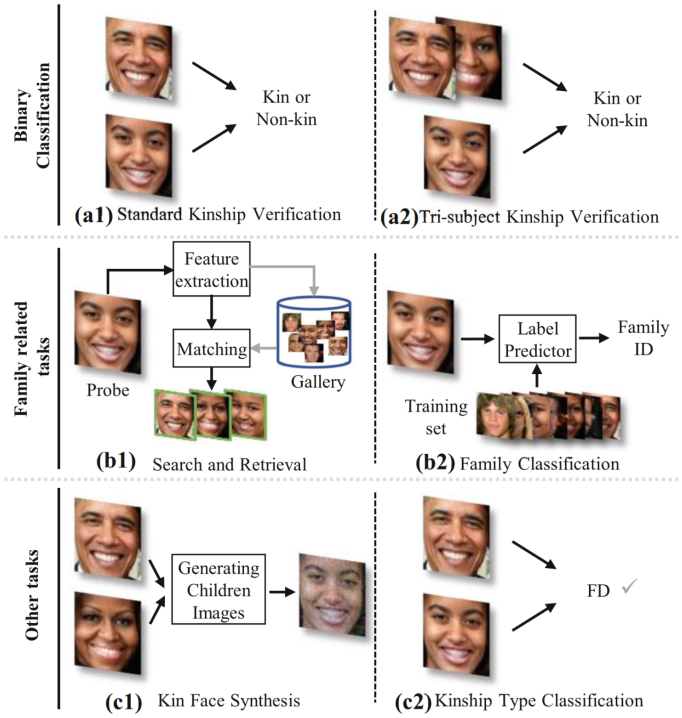


Figure 1. Kinship related tasks, image from Wu et al. (2022)

2.2 The Kinship Verification problem

2.2.1 Problem formulation

The kinship verification problem is defined as given a pair of facial images, to judge whether the two people have a kin-relationship. This task can be further described as a binary classification problem. It can be divided into two steps: feature extraction and kinship classification. The formal expression is: provide a pair of images (x, y), extract high-dimensional feature representations $(f(x), f(y))$ of the two images through a suitable feature extractor, and finally use a classifier to classify whether there is kinship and its confidence.

Anthropological prior knowledge states that similar genes between two close relatives will result in similar faces (Wu et al., 2022). So, this task is based on the similarity judgment between two facial cues. A similar task is the facial verification. The difference is that positive pairs in face verification belong to the same person, while in the case of kinship verification, positive pairs are from two people belonging to a family - which in face verification is a negative pair. But since kinship verification is also based on facial similarity, we can expect that entering the same person's face into the system will get a positive result. At the same time, judging the faces of different ages of the same person can be classified as self-kinship verification.

Furthermore, the goal of kinship verification is to find invariant similar visual features belonging to close relatives. These features often do not come from directly comparing textures of facial images, but some implicit information (Hansen et al., 2020). Compared to facial verification, this problem is more challenging with larger intra-class variation and smaller inter-class variation (Wu et al., 2022). Intra-class variation includes inter-person variation and Intra-person variation. The change of the same person mainly comes from different conditions when the image is collected, including angle, distance, lighting, image quality, and expression changes, etc., which is the same as facial verification. Intra-person variation is mainly due to the fact that positive pairs verified by kinship usually include individuals of different ages and genders, especially parent-child pairs. Larger age differences lead to significant differences in shape and texture. The small inter-class variation comes from some image pairs with similar appearance but no kin-relationship, which also suggests the importance of mining hidden information.

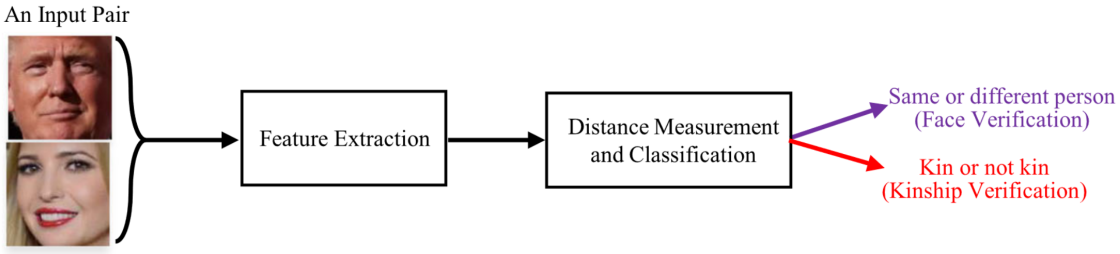


Figure 2. A typical kinship verification system, image from Wu et al. (2022)

2.2.2 Pipeline

A typical kinship verification pipeline usually includes the following steps (Wu et al., 2022):

Preprocessing: Facial landmarks detection and alignment The first step is to extract face images from the input raw images/videos. Firstly, the facial landmarks are extracted. Based on these landmarks, facial images are segmented and aligned. The alignment is to reduce the variation in angle and scale of different facial images. When processing video data, each frame of the raw video is processed separately to obtain a sequence of frames with aligned faces.

Face alignment is another active computer vision sub-research question, which can be classified into 2D (Gao et al., 2010) and 3D alignment (Cao et al., 2013). The review by Wu and Ji (2019) summarizes the research in this field. Murphy-Chutorian and Trivedi (2008) proposed the 68 most commonly used facial landmarks for the first time. MTCNN is one of the most commonly used end-to-end deep learning based approach for facial landmark detection(Zhang et al., 2016). The public software framework OpenFace (Baltrusaitis

et al., 2018) integrates several of the above algorithms and provides a complete set of out-of-the-box solutions.

Feature extraction and distance measurement Having obtained the aligned face images/sequences, the next step is to extract visual kinship features from the input image/sequence pairs. Traditional methods are based on handcrafted features, and neural network methods obtain deep embedding through feature extraction networks. Subsequent chapters will describe the existing research in detail.

After obtaining the feature vectors, the distance between the vectors needs to be calculated using a suitable metric. The traditional methods obtain the distance after transforming the feature vectors through a series of metric learning methods (Li et al., 2016). The deep learning-based methods select positive and negative samples through a certain sampling strategy while training, and calculates the loss in each subsets to guide the neural network to extract appropriate embedding, and finally uses the L2 distance to measure the distance in the embedding space.

Classification After the distance is obtained, traditional methods use clustering algorithms such as K-means, or machine learning models such as support vector machines for classification, and most deep learning methods directly predict the similarly score and make decisions by a pre-defined threshold.

2.2.3 Datasets

Commonly used kin-datasets include image datasets and video datasets. Cornell KinFace dataset (Fang et al., 2010) is the earliest kin-dataset, which contains images collected from the Internet. The UB KinFace dataset (Xia et al., 2011) is the first dataset that emphasizes the parent-child relationship. KinFaceW-I and KinFaceW-II (Lu et al., 2013) are the most commonly used image datasets, which are also collected from the Internet. FIW (Robinson et al., 2018) is the largest and most complex kin image dataset, collected from the wild and contains generation information. The UVA-NEMO Smile dataset (Dibeklioğlu et al., 2012) is the first video kin dataset, which contains video clips of smiles obtained by family members under indoor cooperative conditions. KFVW (Sun et al., 2018) is a kin video dataset collected from the natural environment in the wild, and there are no restrictions on the lighting, posture and other conditions of the shooting object.

2.3 Automated kinship verification approaches

2.3.1 Still Image based methods

Studies using hand-crafted features Traditional methods use hand-crafted features as input to machine learning models.

The very first work on kinship verification start with enumeration features (Fang et al., 2010), which represented the facial features such as eye color, skin color, hair color, geometric characteristics between facial key points (eye, mouth, nose) and face shapes (size of the eyes, mouth or nose). Later, Xia et al. (2012a) has included more descriptive information, such as age, gender, and race. The features are represented with binary features encoded as -1 and $+1$. These features are usually low-dimensional and not comprehensive enough and need additional efforts to manually annotate the samples.

Secondly, methods start to model the salient facial parts such as nose, eyes, mouth (Guo and Wang, 2012). In their work, Goyal and Meenpal (2018) detect the eyes, mouth and nose image area as the salient facial area. Then DAISY descriptor (Tola et al., 2009) is used to extract features and compute the similarity between the input pairs. Kohli et al. (2012) proposed the Differences of Gaussians (DoG) method to extract facial key points and facial landmarks. Besides, Goyal and Meenpal (2018) proposed an edge detection-based kinship feature extraction method. These methods heavily based on the facial shape and usually affected by detection accuracy, variance in facial expression, noise, and face rotation, resulting in low verification accuracy and low noise tolerance under complex conditions.

To solve the aforementioned problems, researchers proposed extracting descriptors for kinship verification. One of the basic descriptor is Local Binary Patterns (LBP)(Freitas Pereira et al., 2012). which is an operator that describes the image's local texture information. The resulting binary code describes the texture characteristics of an image block and is invariant to both rotation and gray-scale conversion.

Based on the basic hand-crafted features, many methods improve the performance in different ways. Pyramid multi-level covariance descriptor (PML-COV) Moujahid and Dornaika (2019) combined the LBP and HOG (Histogram of Oriented Gradients) features extracted from multiple resolutions to form a feature pyramid. Selective Patch-based Dual-Tree Complex Wavelet Transform (SPDTCWT) Goyal and Meenpal (2020) method decomposes the facial image with six wavelet functions and computing the similarity between corresponding patches of an image pair.

The methods mentioned above are based on gray-scale images. To fully use the color

information, Wu et al. (2016) proposed a color-texture feature extraction method to combine color features with texture features. The proposed method first transforms the image into the HSV color space and then extracts features from each color channel. The results suggest that using color information outperform the previous methods using gray-scale images.

Studies using deep features In recent years, CNN-based deep learning methods have shown a strong capability of non-linear representation in the field of Computer Vision. They can learn the effective feature embeddings from the original raw data and avoids limitations of the hand-crafted features. CNN have also used in may sub-areas such as image classification and recognition (Ma et al., 2021) and have also used to represent CNN based deep features on facial images (Li et al., 2016).

Wang et al. (2015) firstly introduced an end-to-end deep learning method for kinship verification. The network inputs are two stacked facial images, and then outputs the final result, which is simple and effective. Further study includes deep CNN (Li et al., 2016), NN-based autoencoders (Dibeklioglu, 2017) and attention structures (Yan and Wang, 2019).

Dibeklioglu (2017) took a pair of kin images as the inputs of dual autoencoders, They made the output of each decoder similar not only to the input facial image but also to its kin facial image. At last, they adopted the encoded features as the kin feature representations.

Yan and Wang (2019) use an attention sub-networks learn the interest areas for the kinship verification task directly from the transformation of the intermediate feature map. They also applied the residual learning idea to retain original information by summing the weighted feature map with the original feature map.

Metric learning was firstly proposed by Xing et al. (2002). For the kinship verification problem, we would need to use a proper distance measure method to compute the distance between an image pair based on extracted features. Ideally, in this metric, the image pairs with kin relations (positive pairs) would have small distances, while those without kin relations (negative pairs) would have large distances.

Zhang12 et al. (2015) proposed the very first Deep Metric Learning approach for kinship verification. While training the network, we use a distance metric to optimize the distance between two input facial images. The typical architecture is Siamese Network. Different from one-stream networks, Siamese networks have two streams that share the same weights

and utilize the distance metric as the loss function to learn an optimal feature space such that positive pairs (pairs with kin relation) have small distances and negative pairs (pairs without kin relation) have large distance.

Li et al. (2016) proposed the Similarity Metric based Convolutional Neural Networks (SMCNN) method into kinship analysis. The inputs of the network are two facial images and a two stream network with sharing weights are used to computer the feature embeddings. Then, the method employed L1-norm to compute the distance of the two output embeddings. Further, the method adds a threshold t to partition the positive samples and negative samples. Network structure are in Figure 3.

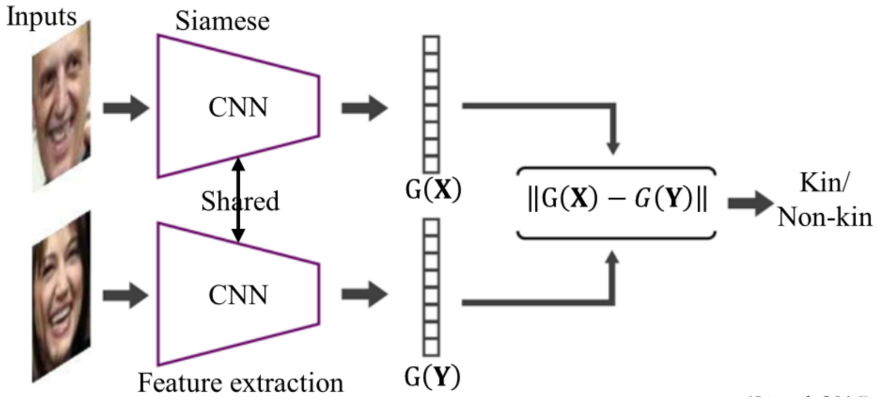


Figure 3. Similarity metric based convolutional neural networks, image from Wu et al. (2022)

2.3.2 Video based methods

A video-based kinship verification system identifies the kin or non-kin relation between subjects present in video sequences containing faces. Compared to still images, facial videos contains additional spatio-temporal information that can be useful for kinship verification.

Dibeklioglu et al. (2013) proposed the first approach that combines of combining appearance and dynamic features to depict kin characteristics. This work hypothesized that people with kin relations might also share similar facial expression dynamic features that could be present in like smiling style. Specifically, Dibeklioglu et al. (2013) extracted the dynamic and facial hand-crafted spatio-temporal features for kinship verification. The work localized 17 facial landmarks to track facial movement and extracted the CLBP-TOP features. Further work by Boutellaa et al. (2017) combined deep features and spatio-temporal features. Experimental results showed that deep features have complementary information compared to the hand-crafted spatio-temporal features. Later, Dibeklioglu (2017) proposed to measure the similarity of kin facial smile videos by matching affective intensity. The work decomposed the smile video into frames and aligned the subsequence according to the smile intensity of the face. The matched sequence pair is input to a dual auto-encoders.

Due to the significant challenges, such as video low quality, blurry frames, dynamic faces Wu et al. (2022), video-based kinship verification has still not reached its full potential. The above works indicate that people with kinship have both similar appearance and smiling expressions, which lights a promising direction for further study.

2.4 The Transformers

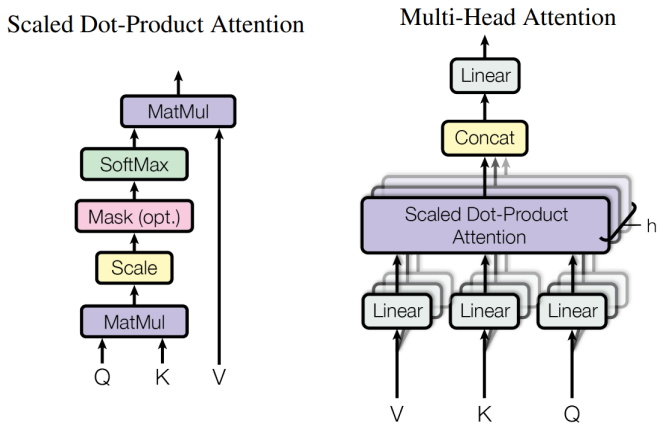
The Transformers by Vaswani et al. (2017) have become a standard neural network for natural language processing in recent years. It has demonstrated exemplary performance on a broad range of language tasks such as text classification, machine translation and question answering. The most popular ones include BERT (Bidirectional Encoder Representations from Transformers) Devlin et al. (2018), GPT (Generative Pre-trained Transformer) Brown et al. (2020), RoBERTa (Robustly Optimized BERT Pre-training) Liu et al. (2019) and T5 (Text-to-Text Transfer Transformer) Raffel et al. (2020).

Transformer architectures are based on a self-attention mechanism that learns the relationships between elements of a sequence Khan et al. (2021). Unlike recurrent networks that process sequence elements recursively and can only attend to short-term context, Transformers can attend to complete sequences thereby learning long-range relationships. Compared to feed-forward and recurrent nets that extensively use attention components, Transformers are based solely on the attention mechanism and have a unique implementation optimized for parallelization. An important feature of these models is their scalability to high-complexity models and large-scale datasets.

Since transformers assume minimal prior knowledge about the structure of the problem compared to convolutional and recurrent nets, they are typically pre-trained using pretext tasks on large-scale unlabelled datasets in a unsupervised manner (Devlin et al., 2018) . Such a pre-training avoids costly manual annotations, thereby encoding highly expressive and generalizable representations that model rich relationships between the entities present in the dataset. The learned representations are then fine-tuned on the downstream tasks in a supervised manner to obtain favorable results.

2.4.1 Self-attention mechanism

The self-attention mechanism is the fundamental part of a Transformer model. It allows capturing long-term dependencies between sequence elements, while traditional recurrent models are hard to encode such relationships. Layer structure can be seen in Figure 4a.



(a) Self-attention layer (b) Multi-head attention layer

Figure 4. Self-attention and multi-head attention (Vaswani et al. (2017))

Self-attention Given a sequence of items, self-attention estimates the relevance of one item to other items. The self-attention mechanism is an integral component of Transformers, which explicitly models the interactions between all entities of a sequence for structured prediction tasks. Basically, a self-attention layer updates each component of a sequence by aggregating global information from the complete input sequence.the input sequnece is first project by 1x1 convolution into Query Q, Key K and Value V, than the output Z is compute through the equation.

$$Z = softmax(\frac{QK^T}{\sqrt{d_q}})V$$

Masked Self-Attention The standard self-attention layer attends to all entities. For the Transformer model Vaswani et al. (2017) which is trained to predict the next entity of the sequence, the self-attention blocks used in the decoder are masked to prevent attending to the subsequent future entities. This is simply done by an element-wise multiplication operation with a mask M. $\odot$ means a Hadamard product.

$$Z = softmax(\frac{QK^T}{\sqrt{d_q}} \odot M)V$$

Multi-Head Attention Multi-Head Attention is designed to encapsulate multiple complex relationships amongst different elements in the sequence. It includes multiple self-attention blocks and each block has its own set of learnable weight matrices, see in Figure 4b.

2.4.2 Model architecture

The architecture of the Transformer model Vaswani et al. (2017) is shown in figure 5. It has an encoder-decoder structure. The encoder consists of six identical blocks, with each block having two sub-layers: a multi-head self-attention network, and a simple position-wise fully connected feed-forward network. Residual connections alongside layer normalization are employed after each block. Note that, different from regular convolutional networks where feature aggregation and feature transformation are simultaneously performed (with a convolution layer followed by a non-linearity), these two steps are decoupled in the Transformer model, self-attention layer only performs aggregation while the feed-forward layer performs transformation. Similar to the encoder, the decoder in the Transformer model comprises six identical blocks. Each decoder block has three sub-layers, first two are similar to the encoder, while the third sub-layer performs multi-head attention on the outputs of the corresponding encoder block. The original Transformer model Vaswani et al. (2017) was trained for the Machine Translation task. The input to the encoder is a sequence of words in one language.

Positional encoding The recurrent neural network is a sequential structure, which inherently contains the position information of words in the sequence. In Transformer, the cyclic structure is completely replaced by self-attention, the order information is lost, and the model have no way to know the relative and absolute position information of each entity in a sequence. Therefore, positional encoding is added to describes the location and position of entities. Each position is assigned a unique representation. It has the same dimensions as the input, and can be learned or pre-defined by sine or cosine functions.

2.4.3 Vision Transformers

Vision transformers (ViTs) (Dosovitskiy et al., 2020) adapts the architecture of (Vaswani et al., 2017) into the field of computer vision, see Figure 6. ViTs have gained significant research attention, and a number of recent approaches have been proposed which build upon ViTs. It is the first work to show how Transformers can 'altogether' replace standard convolutions in deep neural networks on large-scale image datasets.

ViT applied the original Transformer model with minimal changes on a sequence of image 'patches' latent as vectors. Specifically, the work reshapes the images into sequence of flattened 2D patches, and the patches are mapped to a constant dimensions with a trainable linear projection, which is called patch embedding.

In order to maintain the spatial position information between the input image patches, ViT use position encoding vector to the image block embedding. The position encoding of ViT

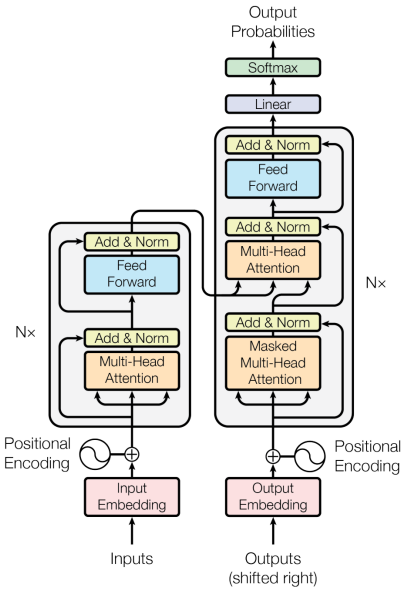


Figure 5. Transformer structure, image from Vaswani et al. (2017)

does not use the 2D position embedding method, but directly uses the 1D learned position embedding variable.

Vision Transformer model is pre-trained on a large proprietary dataset (JFT dataset (Sun et al., 2017) with 300 million images) and then fine-tuned to downstream recognition benchmarks. This is an important step because the CNNs encode prior knowledge about the images (inductive biases like translation equivalence) that reduces the need of data as compared to Transformers which must discover such information from very large-scale data. Compared to the language models (BERT, GPT (Devlin et al., 2018; Brown et al., 2020)) that are pre-trained in an unsupervised manner, ViTs are pretrained with a supervised classification task.

The DeiT (Touvron et al., 2021) is the first work to demonstrate that Transformers can be learned on mid-sized datasets (1.2 million ImageNet examples compared to 300 million images of JFT) in relatively shorter training episodes. Besides using augmentation and regularization procedures common in CNNs, the main contribution of DeiT (Touvron et al., 2021) is a novel native distillation approach for Transformers which uses a CNN as a teacher model (RegNetY-16GF (Radosavovic et al., 2020)) to train the Transformer model. The outputs from the CNN aid the Transformer in efficiently figuring out useful representations for input images. A distillation token is appended with the input patch embedding and the class token. The self-attention layers operate on these tokens to learn their inter-dependencies and outputs the learned class, patch, and distillation tokens. The network is trained with a cross-entropy loss defined on the output class token and a distillation loss to

match the distillation token with the teacher output. The learned representations compare favorably well against top-performing CNN architectures (Touvron et al., 2021) and also generalize well for a number of downstream recognition tasks.

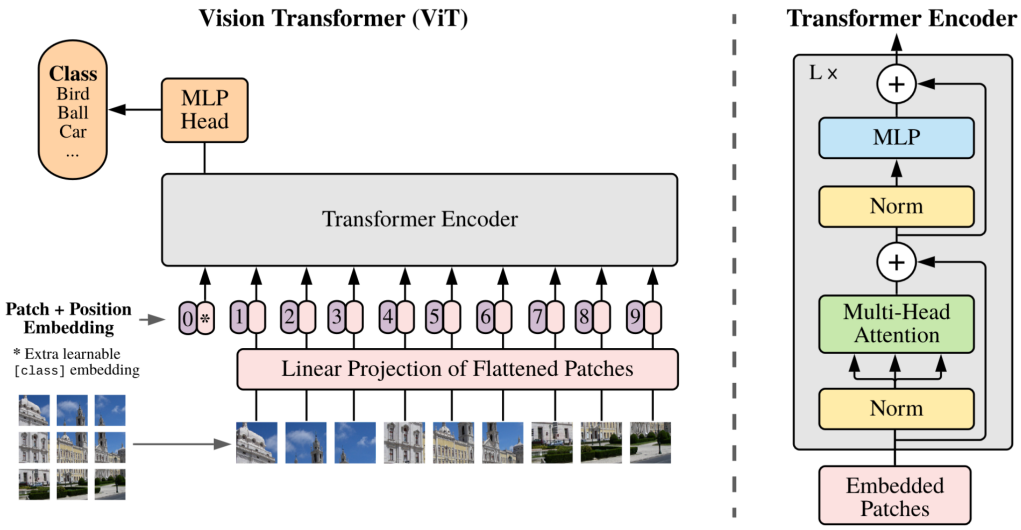


Figure 6. Vision Transformer structure, image from Dosovitskiy et al. (2020)

2.4.4 Transformers on face-related tasks

Vision Transformers have already been adapted to face-related tasks. Zhong and Deng (2021) apply transformer models for face recognition. Jacob and Stenger (2021) apply transformer models for facial action unit detection. The work proposed a transformer encoder architecture to capture the relationships between different facial action units for the wide range of facial expressions. Li et al. (2021a) translated the facial images into sequences of visual words and perform facial expression recognition.

TransFER (Xue et al., 2021) characterizes the relations between different facial parts adaptively, introducing a Multi-head Self-Attention Dropping (MSAD) to randomly remove self-attention modules, forcing models to learn rich relations between different local patches. It introduces introduce Multi-Attention Dropping (MAD) that aims to remove the attention map, pushing the model to extract comprehensive local information from every facial part except the most discriminative parts.

Wang and Wang (2021) introduced a progressive multi-scale vision transformer (PMVT) to capture the complex relationships among different facial action units (AUs) for a wide range of expressions. It is capable of encoding facial regions with adaptive receptive fields, facilitating the representation of different AU flexibly. s VidFace Gan et al. (2021) explores human face super-resolution tasks with ViTs. The work aimed to capitalize on the contextual modeling ability of the attention mechanism to harness all the spatial,

temporal, and facial prior information of given face video snapshots. It particularly uses facial landmark ground truth to regularize the position encoding in the transformer.

2.4.5 Transformers on video analysis

Video Analysis Early works that perfomed video analysis used hand-crafted features to encode appearance and motion information (Borges et al., 2013). The success of AlexNet on ImageNet (Krizhevsky et al., 2012) led to the adaption of 2D CNNs for video as two-stream networks (Simonyan and Zisserman, 2014). These models processed RGB frames and optical flow frames independently before fusing them at the end. Availability of larger video classification datasets such as Kinetics (Kay et al., 2017) subsequently facilitated the training of spatio-temporal 3D CNNs (Tran et al., 2015). However, 3D CNN has more parameters and require significantly more computation than 2D.

Video Vision Transformer Recently, many transformer-based models for video analysis are proposed (Bertasius et al. (2021); Girdhar et al. (2019); Liu et al. (2022); Arnab et al. (2021)). ViViT by Arnab et al. (2021) is one of the state-of-the-art extensions of ViT for videos. To efficiently handle the large number of tokens that may be encountered in videos, this paper proposes several methods to decompose the model along spatial and temporal dimensions to improve efficiency and scalability. Furthermore, in order to efficiently train the model on a smaller dataset, this paper shows how to tune the model by training and utilizing pre-trained image models.

Like the transformers that need to map image patches to token sequences, ViViT considers two simple methods for mapping videos to token, see figure 7.

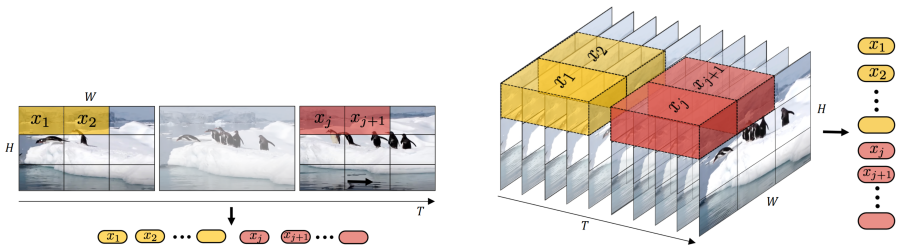


Figure 7. Uniform frame sampling(left), Tubelet embedding(right), image from Arnab et al. (2021)

- **Uniform frame sampling:** Tokenize the input video by sampling the frames uniformly from the input video clip, embed each 2D frame independently using the same method as ViT, and concatenate all these tokens together. Intuitively, this process can be viewed as simply building a large 2D image to tokenize after ViT.
- **Tubelet embedding:** Extract non-overlapping, spatial-temporal tubes from the input

videos and linearly project them into multi-dimensional space. This method is an extension of the ViT embedding to 3D, corresponding to a 3D convolution. Tokens are extracted from the time, height, and width multidimensional space. Intuitively, this approach fuses spatial-temporal information during the token, contrast to unified frame sampling in which temporal information from different frames is fused by the transformer.

Further, the paper proposes several transformer-based architectures for videos. It starts with a simple extension of ViT that models pairwise interactions between all spatio-temporal tokens, then develops more efficient variants. Each architectures can be seen in Figure 8.

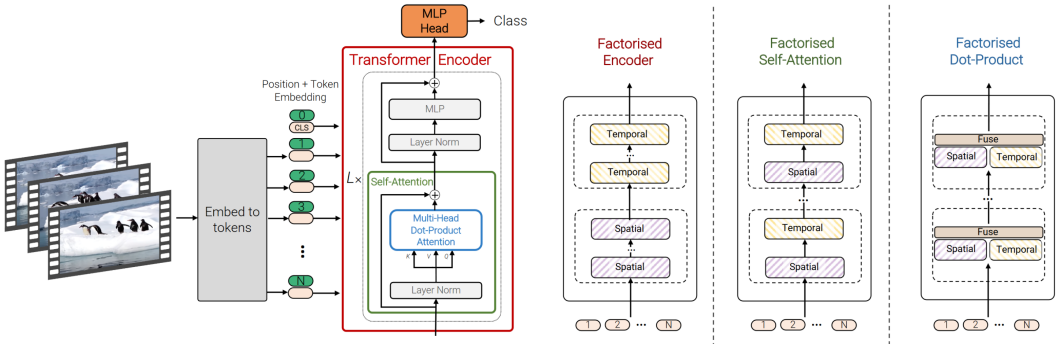


Figure 8. ViViT structure, image from Arnab et al. (2021)

- **Factorised encoder:** Consists of two concatenated transformer encoders: the first transformer encoder models the interactions between tokens extracted from the same temporal index. The second transformer simulates the interaction between time steps. Therefore, it corresponds to a late fusion of spatial and temporal information. And the initial spatial encoder is the same as the one used for image classification. Therefore, it is similar to CNN architectures, which first extract features from each frame and then fuse them into a final representation before classification.
- **Factorised self-attention:** Instead of computing multi-head self-attention across all token pairs, decompose the operation to first compute self-attention only spatially (on all tokens extracted from the same temporal index), and then temporally (on from the same spatial index)
- **Factorised dot-product attention:** factorise the multi-head dot-product attention operation, compute attention weights for each token separately over the spatial and temporal-dimensions using different heads.

ViT has been shown to be effective only when trained on large-scale datasets, since transformers lack some of the inductive preferences of convolutional networks. However, even the largest video datasets, such as Kinetics Kay et al. (2017), have much fewer labeled

examples than image datasets. Therefore, training large models from scratch to high accuracy is extremely challenging. To solve this problem and make training more efficient, Arnab et al. (2021) initializes the video model from a pre-trained image model.

For positional embeddings, at initialization, all tokens with the same spatial index have the same embedding as the image model. For embedding weights, following the common approach for initializing 3D convolutional filters from 2D filters, the paper tutor inflates the 3D filters by replicating the filters along the temporal dimension and averaging them. For Model 3, Arnab et al. (2021) initializes the spatial MSA module from the pre-trained module, and initialises all weights of the temporal MSA with zeroes.

2.5 Visual Representation Learning Methods

To obtain meaningful visual representations for kinship verification, we use a range of methods. Our approach can be divided into two steps: obtaining initial weights via one of the following pre-training methods, and fine-tuning on downstream tasks based on a contrastive learning loss.

In this section, we introduce the pre-training methods we referred to in our experiments, and also the contrastive learning method we are using in the following steps.

2.5.1 Pretraining methods

Pre-training allows the model to learn general features that are useful for the specific task, and to have a good starting point for fine-tuning for the specific task. It can also help to reduce the amount of labeled data that is required for the target task, as the model is able to make use of the knowledge it has learned from the pre-training phase. This can be particularly useful when labeled data is scarce or expensive to obtain. In our case, we have an extremely small dataset compared to the pre-training set.

Pretraining on Large Datasets

One of the ways to do this is to pretrain a network on a large dataset of images, such as ImageNet, and then fine-tune the network on a smaller dataset of images specific to the task at hand. This is called transfer learning. In transfer learning, the first few layers of the pre-trained model are fine-tuned on the new dataset with a smaller learning rate, while the deeper layers are kept fixed. The reason behind this is that the lower-level feature detectors, such as edges and textures, learned during the pre-training on a large dataset are highly general, and can be directly applied to other datasets, while the higher-level feature detectors, such as object parts, are task-specific and require further fine-tuning.

Self-supervised pre-training based on mask autoencoder

Another way to do pretraining on computer vision tasks is by using unsupervised pre-training method such as autoencoder or Generative Pre-training approach such as GPT-2, where the model is trained to recreate the input or learn to generate new samples that looks similar to the input. This is a method for training deep neural networks to learn visual representations from data without the need for explicit supervision. By learning to make these predictions, the neural network is able to learn useful feature representations of the data.

An autoencoder is typically composed of two parts: an encoder network that maps the input to a lower-dimensional representation, called the bottleneck or latent representation, and a decoder network that maps the bottleneck representation back to the original input. The encoder network can be thought of as a feature extractor, while the decoder network can be thought of as a feature generator.

To pretrain a neural network using an autoencoder, the network is first trained on a large dataset of unlabelled images to minimize the reconstruction loss between the input and the output of the autoencoder. Once the autoencoder is trained, the encoder network can be used as a feature extractor for a computer vision task. The encoder network can be further fine-tuned on a smaller dataset of labelled images, or it can be used to extract features that can be input to a classifier or a regressor.

VideoMAE (Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training) by Tong et al. (2022) is a paper that presents a self-supervised learning method for training a Video ViT network to learn visual representations from video data. They proposed a novel way of masking out random cubes of the input video sequences and training the autoencoder to reconstruct the original video from the remaining cubes. This allows the autoencoder to learn to attend to different parts of the frame and combine them to form a global image representation.

We used this method in our experiments and further explain it in section **??**.

Pre-training based on expression alignment

Dibeklioglu (2017) proposes to learn efficient kinship representations by modeling a visual mapping that can transform an object's facial appearance into a form that closely resembles the faces of his/her relatives, while reducing the similarity pattern from a non-kin person.

To obtain this, Dibeklioglu (2017) uses similar expressions from the subject pairs for pre-taining. It proposed a method to compute smile expression similarity. After getting the

alignment expression sequences, it feeds similar expression pairs into an encoder-decoder network. To reveal the facial resemblance patterns between kin pairs, these encoder-decoder networks are trained in a way that each network outputs a face image that is similar to its input's kin pair while minimizing its resemblance to non-kins.

Dibeklioglu (2017) argues that optimizing facial representations in visual space can be thought of as an extension of unsupervised pre-training. Unsupervised pre-training effectively constrains the form of prediction functions by learning sparse representations. The proposed method constrains the latent representations so that it can capture blurred visual kinship pairs while discarding similar patterns between non-kin. Therefore, it can theoretically be claimed that the proposed method will provide stronger regularization by simplifying the kinship validation model (during pre-training) in a different but related space. This study is the first exploration of visual transformation-assisted deep representation learning for kinship verification.

2.5.2 Contrastive learning

Contrastive learning is a machine learning technique that involves training a model to differentiate between two or more distinct classes or concepts. This is typically done through the use of a contrastive loss function, which measures the similarity between the output of the model and the true label and adjusts the model's weights accordingly. It is often used in unsupervised learning scenarios, where there are no explicit training labels available. In these cases, the model must learn to distinguish between different classes or concepts based on the inherent structure of the data. For example, a contrastive learning model might be trained to distinguish between different types of animals based on images of their faces. The model would be shown a series of images of different animals and would be trained to predict the correct label for each image based on the features and characteristics present in the image. Through this process, the model learns to differentiate between different animals, and becomes better at classifying new, unseen images based on this learned knowledge.

2.5.3 Loss functions

Contrastive loss The contrastive loss function is defined as:

$$L = (1 - y) \times D^2 + y \times max(margin - D, 0)^2$$

where D is the distance between the two examples, y is a binary label indicating whether the examples are similar (y = 1) or dissimilar (y = 0), and margin is a hyperparameter that determines the minimum distance required between similar and dissimilar examples.

In practice, the distance between examples is often measured using a distance metric such as the Euclidean distance or the cosine similarity. The contrastive loss function can then be minimized using an optimization algorithm such as stochastic gradient descent (SGD) or Adam.

Contrastive loss is often used in combination with a neural network architecture known as a Siamese network, which consists of two identical sub-networks that are trained to process the input examples and generate feature vectors that are then used to compute the distance between the examples. By minimizing the contrastive loss, the Siamese network can learn meaningful feature representations that can be used to distinguish between similar and dissimilar examples.

Triplet loss Triplet loss is a type of loss function that is used in training deep neural networks for tasks such as image classification and face recognition. It is based on the idea of maximizing the distance between examples from the same class (i.e., positive examples) and minimizing the distance between examples from different classes (i.e., negative examples).

The triplet loss function is defined as:

$$L = max(d(A, P) - d(A, N) + margin, 0)$$

where A is an anchor example, P is a positive example (i.e., an example from the same class as the anchor), N is a negative example (i.e., an example from a different class), d is a distance function (such as the Euclidean distance or the cosine similarity), and margin is a hyperparameter that determines the minimum required distance between positive and negative examples.

To optimize the triplet loss function, a neural network is trained to generate feature vectors for the anchor, positive, and negative examples. These feature vectors are then used to compute the distances between the examples, and the triplet loss is minimized using an optimization algorithm.

Triplet loss is often used in combination with a neural network architecture known as a triplet network, which consists of three sub-networks that are trained to process the anchor, positive, and negative examples and generate feature vectors that are then used to compute the triplet loss. By minimizing the triplet loss, the triplet network is able to learn

meaningful feature representations that can be used to distinguish between examples from different classes.

InfoNCE loss Information Noise Contrastive Estimation is a method for training deep neural networks to learn meaningful representations of data. It is based on the idea of using a noise contrastive loss function to maximize the difference between the distribution of the representations of the data and the distribution of the representations of noise.

The input to the InfoNCE loss function consists of a set of data points (such as images or text) and a set of labels indicating the class of each data point. The data points and labels are used to compute the probability of generating the desired output (i.e., the label) given the input data. In addition, the InfoNCE loss function also requires a set of noise data points, which are sampled from a fixed noise distribution. These noise data points are used to compute the probability of generating the desired output given the noise data.

The noise contrastive loss function is defined as:

$$L = -\log(p(D|X)) + \sum_i [\log(p(D|X_i))]$$

where X is the input data, D is the desired output (e.g., a label indicating the class of the input data), $p(D|X)$ is the probability of generating the desired output given the input data, and $p(D|X_i)$ is the probability of generating the desired output given a set of noise data points X_i.

By minimizing the InfoNCE loss function, the neural network is able to learn feature representations that are more likely to generate the desired output given the input data, and less likely to generate the desired output given the noise data. This helps the neural network to learn more meaningful and discriminative representations of the data.

3. Methodology

In this chapter, we introduce the methodologies proposed to answer the RQs. The first part is about data preparation and preprocessing, and then shows the model and experiment design.

3.1 Data Preparation and Evaluation Metrics

3.1.1 Dataset overview

We have used the UvA-NEMO dataset Dibeklioğlu et al. (2012). It is a dataset of facial images and videos of individuals smiling. It was developed by the researchers at University of Amsterdam (UvA) as part of the NEMO (Naturalistic Emotion and Motivation Observatory) project, which aims to study the emotions and motivations of individuals in naturalistic settings. It is a large-scale smile database that has 1240 smile videos (597 spontaneous and 643 posed) from 400 subjects. The dataset includes images and videos of individuals smiling naturally, as well as images and videos of individuals forced to smile. The dataset also includes annotations for demographic information about the individuals. The UvA-NEMO Smile Database is intended for use in research on facial expression recognition, emotion analysis, kinship verification, and related topics. It has been used in a number of studies on the recognition and interpretation of facial expressions of emotion.

The data set includes the following types of kinship subsets, as shown in the Table 1

	Spontaneous		Posed	
Relation	Subject	Video	Subject	Video
S-S	7	22	9	32
B-B	7	15	6	13
S-B	12	32	10	34
M-D	16	57	20	76
M-S	12	36	14	46
F-D	9	28	9	30
F-S	12	38	19	56
All	75	228	87	287

Table 1. UvA-NEMO dataset

Most of the videos are clips of 1-2 seconds at 50 fps and the average total number of frames is around 70 frames. Each subject has 2-4 videos, including natural and unnatural smiles respectively. Multiple subjects come from the same family, with different kin-relationships.

3.1.2 Data preparation

We first extract and align facial image sequences from the raw video as explained in section 2.2.2. We use the out of box tool-set OpenFace Baltrusaitis et al. (2018) to process the whole dataset. Each extracted video clips contains 100-200 frames with a size of 112 $\times$ 112 pixels. The toolbox also provides functions to identify the facial action units (AUs) and their intensity. While segmenting the face image, we identify and record the AU12 value representing the smiling action to facilitate subsequent experiments. Further, only videos of natural smiles are used for every experimnent, since it is one of the most frequently shown facial expressions, refering to Dibeklioglu (2017)

3.1.3 Dataset splitting and Evaluation metrics

Previous works on this dataset by Boutellaa et al. (2017) split the dataset by kinship subcategories and trained models for each subset. Each time videos of a test pair are separated and the system is trained using leave-one-subject-pair-out cross-validation on the remaining pairs. random pairs that do not have a kin-relation are used as negative samples. These random pairs are specifically constructed for each subset. For example, a positive example of a father-son relationship will have a different son subject of similar age (age range $\pm$5 years) as a negative example.

Due to the large amount of data required for training Vision Transformers and the long training time, leave-one-subject-pair-out cross-validation is not suitable for our experiments. Therefore, for experiments presented in the sections 3.2.2,3.2.3 and 3.2.4, we randomly split the data set into training, testing, and verification sets by the ratio of 0.8, 0.1, and 0.1. We ensured that the subjects of the same family would not appear repeatedly in different data splits while splitting the data set. We randomly generate the splits three times and keep them for all experiments. All experiments were repeated three times with different splits to ensure a fair comparison.

In the labels, since multiple subjects belonging to one family appear as different kin relationship pairs, directly randomly shuffling the kinship pairs and splitting the training, testing, and validation sets may result in the same subject appearing in both the testing and training sets. For example, subject 001 and 002 are in the training set as a father-son pair, and 001 and 003 may appear in the test set as a father-daughter pair, so subject 001 may appear in both sets at the same time. To avoid this, we first cluster the subjects belonging to the same family and ensured that the subjects of the same family would not appear repeatedly in different data splits while splitting the data set.

3.1.4 Expression matching

In our experiments in 3.2.4, referring to Dibeklioglu (2017), pairs of sequences with very similar facial expressions were used as input to obtain enhanced facial representations. As part of data preparation, we performed expression alignment. Dibeklioglu (2017) matches faces with similar expressions based on handcrafted features designed according to shape variation by tracking facial key points. Using the extracted features, each "2m + 1" frame sub-sequence of the input video (obtained by a sliding window) is matched to its "2m + 1" frame sub-sequence of the paired video. Based on the results by Dibeklioglu (2017), we choose m=2 as the temporal matching width. In our work, we discarded the complex handcrafted features and directly used the AU12 intensity obtained in section 3.1.2, and used the AU12 intensity of the 2m+1 frames as the feature for expression matching, and successfully obtained similar facial expression pairs. Among them, as outliers, frames in which individuals do not smile and corresponding AU 12 intensity values are zero are discarded.

3.2 Model Design and Experiments

In this section, we present our model design and several experiments used to answer the research questions. Firstly we aim to answer sub-research questions 1 and 2 by conducting two sets of experiments. The first experiment, following Boutellaa et al. (2017), is a simple approach that used an off-the-shelf pre-trained network to extract features from images and trained a Support Vector Machine (SVM) model for binary classification kin or not-kin. The second experiment, inspired by Zhang12 et al. (2015), attempts to train a deep learning model for kinship classification. In these two sets of experiments, we compared the performance of using spatial feature and spatial-temporal features, convolutional methods, and transformer methods for kinship verification.

In the following two experiments, we focus on further using video clips and transformer models for kinship verification. We employed a Video Vision Transformer based Siamese network and used contrastive learning methods for training. We also introduced using a self-supervised pre-training method to initialize the model weights. We further performed experiments using aligned expression sequences for training to improve our result.

3.2.1 Off the shelf deep feature for binary kinship classification

Pre-trained deep learning models provides a good prior distribution that can be used to extract facial feature vectors. Based on Zhang12 et al. (2015), we try to directly use the pre-trained model for feature extraction from facial videos, and then train a simple SVM classifier for classification.

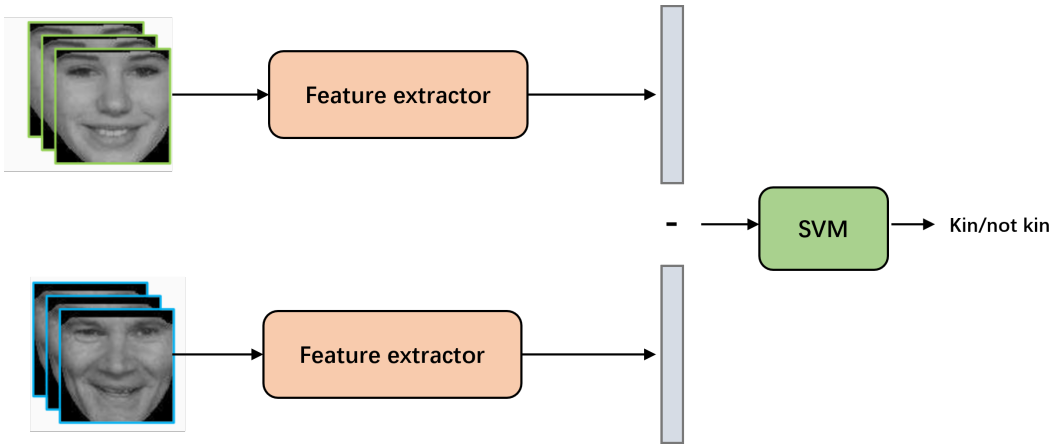


Figure 9. Off the shelf deep feature for binary kinship classification

We experimented with three pre-trained models respectively for video extraction. They are ResNet50, ViT and ViViT. ResNet50 and ViT are pre-trained on imagenet, while ViViT is pre-trained on Kinetic-400.

The first two models are used to extract image features. Referring to Zhang12 et al. (2015), we calculate a feature vector for each frame of the video and use the mean of these vectors as the feature vector of this video. ViViT directly extracts the spatio-temporal features of the video and obtains a vector as the feature vector of this video. As we can see, the first two models only extract spatial features, and ViViT extracts spatial-temporal features.

Before feeding the features to the SVM, each pair of features has to be transformed into a single feature vector as imposed by the classifier. We have examined various ways for combining a pair of features, such as concatenation and vector distances. We have empirically found that utilizing the normalized absolute difference shows the best performance, as same as Zhang12 et al. (2015). Therefore, in our experiments, a pair of feature vectors $X = x_1, ..., x_d$ and $Y = y_1, ..., y_d$ is represented by the vector $F = f_1, ..., f_d$ where :

$$f_i = \frac{\sum_{j=1}^{d} |x_j - y_j|}{\sum_{j=1}^{d} (x_j + y_j)}$$

3.2.2 Deep learning models for binary kinship classification

In this experiment, we trained a 3D CNN-based architecture named MobileNet-3D and a ViViT respectively as the feature fusion module to process the features extracted by the pre-trained model for the binary classification task of the kinship. The purpose of this experiment is to compare the ability of the convolutional structure and the ViT structure in processing spatio-temporal information for kinship verification. The model is shown in figure 10.

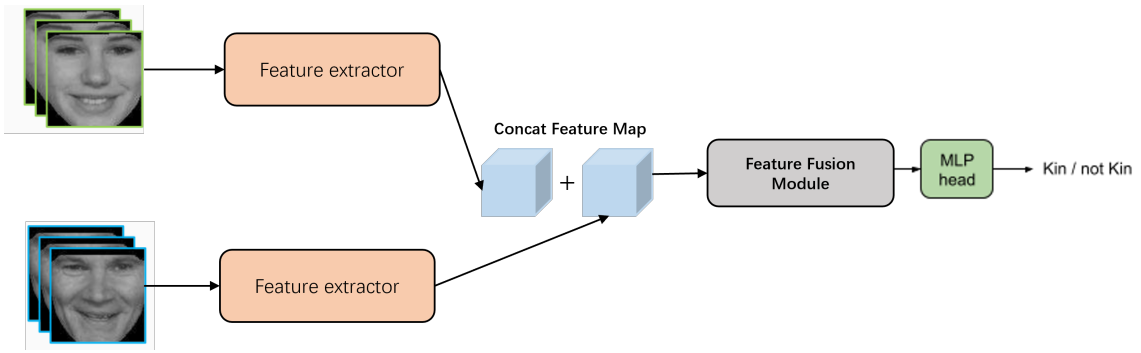


Figure 10. Deep learning model for binary kinship classification

First, we use a pre-trained feature extractor to extract the feature maps of each input frame and stack the obtained feature maps. That is, a 112×112 video clip with n frames will get a $n \times 112 \times 112$ feature map. We extract such a feature map for a pair of input segments, and then concatenate the two feature tensors directly in the time dimension to get a a $2n \times 112 \times 112$ feature map, and use them as the input of the next feature fusion module. The feature fusion module processes the feature map information, and finally outputs a binary classification result through an MLP classification head. The model is trained using cross-entropy loss.

Implementation details For fair comparison, the feature extractors of both experiments are ResNet50 pre-trained on imagenet. Here, we directly remove the fully connected layer of ResNet50, and use the last layer of feature maps as the input of the network to keep the input structure of the MobileNet-3D and ViViT network as feature extractors consistent with the original model. As we have a really small dataset, we use the smallest setting of MobileNet-3D and a ViViT with only 2 layers. Since using a larger time width will make the feature map larger and make it difficult to train the classifier, we choose n=8 as input, that is, uniformly sample 8 frames from the beginning to the end of each video clip to preserve the entire smile. During training, we freeze the feature extractor and only train the feature fusion module and its head.

3.2.3 Video Vision Transformer based Siamese network

As the main contribution of our thesis, We worked on a novel network structure, a Video Vision Transformer based Siamese network on kinship Verification tasks. The overview of the network is shown in figure 11.

Our proposed model has a two-stream pipeline like the Siamese network. The input is a pair of aligned smile face video clips. Firstly, each of the frame sequences is embedded into a 1D patch sequence following the strategy explained in section 2.4.5. Then, a patch and position embedding is added to each of the sequences. the sequences of parent and child are

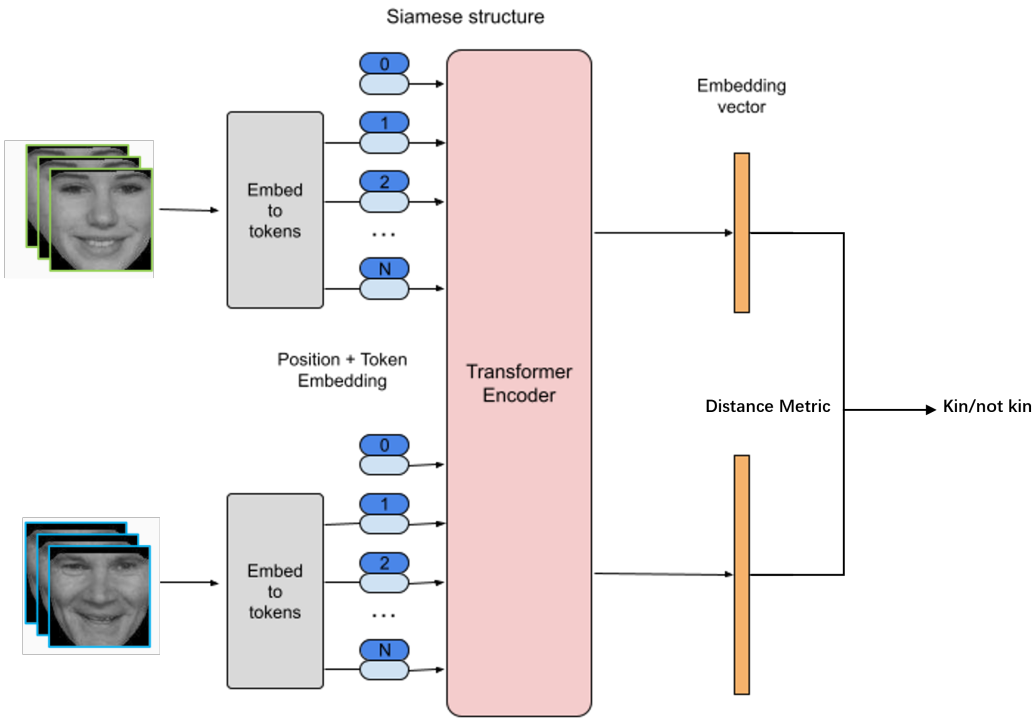


Figure 11. Video vision transformer based Siamese network

then passed through the same transformer encoder separately to get two output embeddings. Finally, the output embeddings will go through an MLP layer for classification to determine whether their have kinship or not.

Embedding video clips We follow the Tubelet embedding explained in section 2.4.5. We consider the video clips with a dimension of $T \times H \times W$, for a tubelet of $t \times h \times w$ dimension, $n_t = \frac{T}{t}$,$n_h = \frac{H}{h}$,$n_w = \frac{W}{w}$ tokens are extracted. The video clips thus is projected linearly into a 1D tublet sequences $z \in \mathbb{R}^d$. Intuitively, this tokenisation method well fuses the spatio-temporal information as we want to extract the facial dynamic information.

Positional encoding After that, a learned positional embedding, $p \in \mathbb{R}^{N \times d}$ is added to the tokens to retain positional information.

Self-supervised pre-training The weights officially provided by ViViT are pre-trained on the Kinetics-400 dataset (Kay et al., 2017). To better fit the weights to our face dataset, we follow the strategy developed by (Tong et al., 2022), using self-supervised pre-training to initialize the model's weights. As mentioned in **??**, we added a small decoder to vivit to form an autoencoder structure, masking out random tubes of the input frame and training

the autoencoder to reconstruct the original frame from the remaining tubes. This allows the model to learn to focus on different parts of the frame and combine them to form a global image representation. Following Tong et al. (2022), we used a masking ratio of 90%, and used MSE loss for trainning. In this way, we initialized the weight of our model for further fine-tuning.

Experiments to compare the impact of loss functions In the experiments, we tested the training with triplet and contrastive loss functions, introduced in Section 2.5.3. We set the distance parameters in the two losses to 1. Our experiments use the training set for pre-training and contrastive learning for fine-tuning. Then we observe observe the performance of the model on the validation set, determine an optimal split weight, and then use this weight to calculate the result on the test set as our final result.

3.2.4 Expression alignment based pre-training

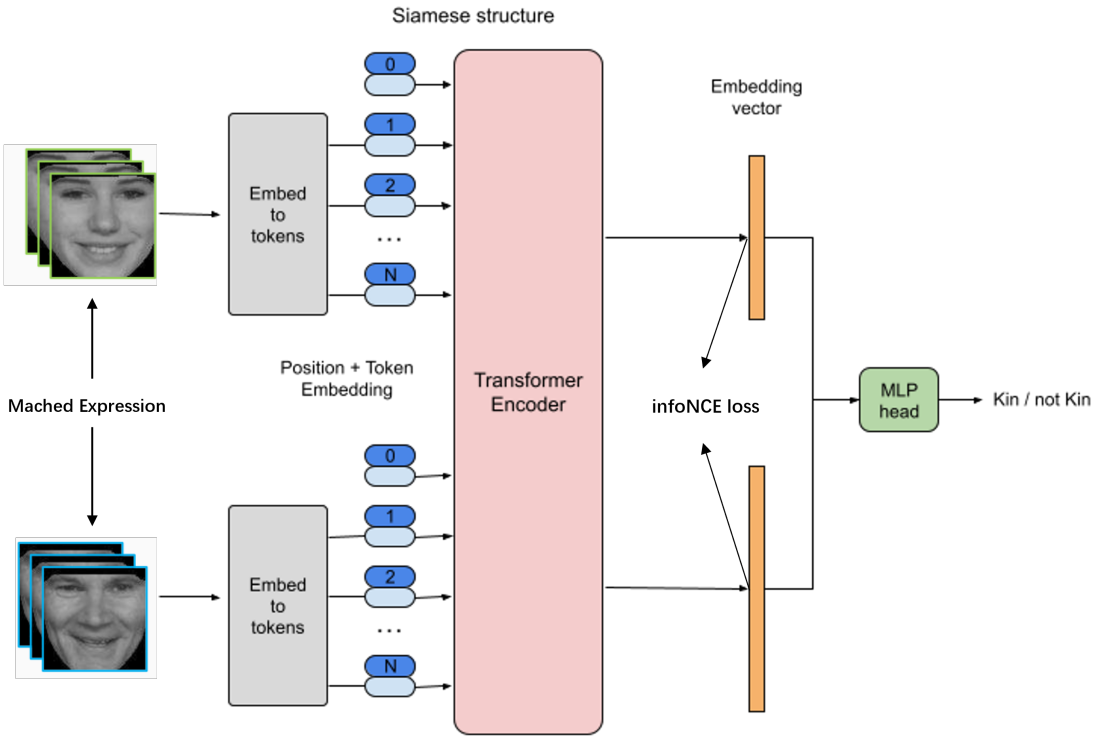


Figure 12. Video Vision Transformer based Siamese network trained by InfoNCE loss

When performing the aforementioned experiments in Section 3.2.3, we face many challenges. First, our data set is too small, and ViT has a strong fitting ability. Although we have used the entire data set for training, the problem of overfitting is still extremely serious. Second, the intra-class distance for this problem is large, but the inter-class distance is small. Specifically, unrelated people may have similar smiles, and related people may have large differences in appearance due to age differences. Combining the above two

factors, if we want the model to specifically learn the similarity of smiles between relatives, we need a more explicit guidance model for learning. So in this section, we introduce a pre-training method based on expression alignment, see figure 12. This method is inspired by Dibeklioglu (2017) which has been introduced in Section 2.5.1.

The statistics of matched expression pairs we obtained in preprocessing introduced in Section 3.1.2 are shown in the table 2.

Subject	Video	Matched Sequence
95	1031	279,578

Table 2. number of matched expression sequence

In this experimental setting, a pair of matched facial image sequences of two subjects with kinship is taken as positive examples, while negative examples consist of (i) non-matching facial image sequences of two subjects with kinship, (ii) matched sequences without kinship, and (iii) non-matched sequences with no kinship as shown in Table 3. Such a design explicitly emphasizes similarity patterns of visual kinship while discarding similarity patterns observed between unrelated persons.

label	kin	non-kin
matched	1	0
not-matched	0	0

Table 3. Assigning class label 0 or 1 based on the matched / not-matched expressions and kin / non-kin relations

We use infoNCE loss for training, which has been introduced in Section 2.5.3, the label of the positive example is set to 1 and the negative example is 0.

For the infoNCE losses does not directive targeting the L2 distance between the inputs, a simple fully connected layer is added after the above parts to perform the classification. It takes the two output embedding and computes a score d representing their distance for kinship classification. We trained the small classification head after the pre-training to determine the kin-relations.

4. Results

you need to explain each table in detail. For example in 4.1 you need to mention that you report results for 7 kinship pair types for each method. You should compare your performances with others and among each other (e.g. using ResNet50 features and ViViT features gave similar performance and are slightly higher than ViT. These methods outperform some of the available approaches SMCNN, VGG-face + Temporal but still VTCL outperforms them).

4.1 Off-the-self feature extractor

Methods	M-D	M-S	F-D	F-S	S-S	B-B	S-B	Mean
SMCNN	83.58	81.46	85.15	84.81	84.64	86.43	85.84	84.56
VGG-face + Temporal	91.23	90.49	93.10	88.30	88.93	94.74	90.07	90.98
VGG-face	90.24	85.69	89.70	92.69	88.92	92.82	88.47	89.79
VTCL	93.64	92.24	93.83	93.35	94.18	95.71	92.58	93.65
ResNet50	90.24	92.82	93.24	94.16	88.92	89.79	90.69	91.24
Vit	91.16	92.10	92.69	93.47	88.47	89.70	90.24	91.09
Vivit	91.24	92.69	94.16	95.79	85.69	87.82	89.70	91.25

Table 4. Accuracy (%) of different methods on UvA-NEMO database

Table 4 shows the kinship verification accuracy of the experiments using off-the-shelf feature extractors and training SVM as proposed in Section 3.2.1 on the UvA-NEMO dataset.

The table shows the result of 7 kinship pair types and their mean value for each method. The upper part of the table shows the results of several previous methods on UvA-NEMO, where the data comes from Boutellaa et al. (2017). The lower part of the table shows the result of our experiments in Section 3.2.1 using pretrained ResNet50, ViT and ViViT respectively.

We can find that using ResNet50 features and ViViT features gave similar performance and are slightly higher than ViT. These methods outperform some of the available approaches SMCNN, VGG-face + Temporal but still VTCL (Dibeklioglu, 2017) outperforms them.

Amoung the methods mentioned in the table, SMCNN, VGG-face, ResNet50 and Vit are

spatial feature extractors, VGG-face + Temporal, VTCL, and vivit are spatial-temporal feature extractors. We can find that the spatial-temporal feature extractors outperform their corresponding spatial feature extractors, for example the improvement of the VGG-face + Temporal group compared to the VGG-face group. When look between specific kinship pair types, we can find that groups with large age differences have improved the most.

4.2 Deep learning models for classification

Methods	MobileNet-3d	ViViT-2
Accuracy	54.91	57.72

Table 5. Accuracy (%) with different feature fusion module

Table shows the accuracy of binary kinship classification on the test set of UvA-NEMO. The two results are based on using Resnet50 features and training 3d-MobileNet & ViViT-2 as feature fusion modules respectively. We use the smallest configuration of 3d-MobileNet and a ViViT with two attention layers called ViViT-2 here.

Both CNN based approach and vision transformer based approach give very poor performances compared to training SVM with the features obtained by off-the-shelf feature extractors. We are going to further discuss it in Section 5.5.

4.3 Video Vision Transformer based Siamese network

Methods	Triplet loss	Contrastive loss
Accuracy	59.65	61.89

Table 6. Accuracy (%) trained by different loss function

Table 6 shows the kinship verification accuracy on the test set when training the network proposed in section 3.2.3 with different losses. The model is initialized using a self-supervised pre-training method.

Methods	train from scratch	Kinetics-400	self-supervised	+ expression alignment
Accuracy	51.20	58.60	61.89	67.66

Table 7. Accuracy (%) trained by different loss function

Table 7 demonstrates the accuracy of kinship verification on the test set when the model proposed by Section3.2.3 is initialized in different ways and fine-tuned.

It shows the result respectively when the model is trained from scratch, using pre-trained

weights from Kinetics-400, or self-supervised pre-trained weights. The right column also shows the result when the model is further trained on aligned expression segments under self-supervised pre-training weights in 3.2.4.

Training from scratch yield almost chance performance. Using the weights of the model pre-trained with Kinetics-400 gives an accuracy of 58.60, which is slightly worse than the self-supervised model which gives an accuracy of 61.89. And using InfoNCE loss and trained on aligned smiling faces gain the best accuracy of 67.66.

5. Discussion

5.1 Spatial feature and Spatial-temporal feature

We answer sub-research question 1 here: How does using spatial-temporal features compare to only using spatial features for video-based kinship verification?

Results of our experiments on using the features extracted by off-the-shelf extractors and training SVMs (reported in Table 4) show that the spatial-temporal features extracted by ViViT are slightly better than the texture features extracted by ResNet and ViT. When the difference between subjects is large (e.g. different age and gender), the spatio-temporal features are more critical. This is consistent with what has been seen in previous studies, such as the improvement of the VGG-face + Temporal group compared to the VGG-face group in Boutellaa et al. (2017).

5.2 Vison Transformer and Convolutional network

We answer sub-research question 2 here: Can vision transformers learn better representations for kinship verification than convolutional neural networks?

Two sets of experiments have been used to compare the performance of convolutional neural network based approach and vision transformer based approach. First, we compare the performance of using pretrained CNN and ViT as feature extractors (as given in Table 4). It can be found that the features extracted by the ViT group provide slightly worse results compared to the ones extracted by ResNet50. The improvement of ViT compared with ViViT can reflect the advantages of the ViT method in extracting context information. The results of concatenating deep features and training deep models for binary kinship verification experiments (table 4.2) also shows that compared with 3d-MobileNet, using ViViT has a slight advantage in extracting and summarizing spatio-temporal features. A previous work by Raghu et al. (2021) studied the difference between features extracted by ViTs and CNNs. One of its conclusions contradicts with part of our result, that ViT retains more spatial information than ResNet. And there currently no works study the difference between features extracted by ViViTs and 3d-CNNs.

5.3 Loss functions for contrastive learning

We answer sub-research question 3 here: How does the choice of loss function for contrastive learning influence the performance of the kinship verification model?

In our third set of experiments, we tried two different kinds of loss functions namely triplet loss and contrastive loss for training. It can be seen from table 6 that the accuracy of contrastive loss is slightly higher. We believe that in the a binary classification task, contrastive loss is more intuitive to push the sample distance d to the two absolute values 0 and 1, while triplet loss pays more attention to the relative distance between sample distances, that is, $|d1 - d2|$, and does not pay attention to the absolute distance, which is not good for binary classification problems that need to determine a classification threshold. So, the contrastive loss is better for training in our case. There are currently no articles that systematically compare the performance of the two losses under different tasks

5.4 Impact of pre-training methods on the kinship verification performance

We answer sub-research question 4 here: Can pre-training methods enhance the performance of the kinship verification model?

The results of experiments training with different pre-training methods are shown in table 7. We can clearly see that the training method from scratch hardly learns useful representations, and pre-training the models with Kinetic-400 in a supervised manner yields worse results compared to self-supervised pre-training. This can be potentially caused by the large difference between UvA-NEMO dataset and kinetic action recognition dataset is too large. Kinetic-400 mainly contains some common outdoor scenes, and we are facial videos. Self-supervised learning can better learn the feature representation related to this data set. When training with matched expression pairs, the infoNCE loss functions better guides the model to learn discriminative representations.

5.5 Vision Transformer for Kinship Verification

We answer the main research question here: How does Vision Transformer-based siamese-network perform on video-based kinship verification task?

We can find that compared to the first method using off-the-self features and simple SVM classifier, the accuracy of our proposed models is quite low. First of all, because the training time of the ViT is too long compared with the conventional model, subsequent experiments have to use different protocols, as described in section 3.1.3. So there is no comparison between the two. Secondly, the training of ViT requires a lot of data, and our data set is too small, that is, there are very few subjects available. There are only 10-20 samples for each category corresponding to 100 videos. Although through combination, there are many matched pairs available for training, the variation in the dataset is still

small. Although we used different pre-training methods and we also proposed expression matching methods which slightly alleviated the problem of insufficient training data by splitting a large number of aligned expression sequences, but still did not solve the problem of the small number of experimental subjects.

At the same time, we also want to criticize the protocol used in the first experiment using off-the-self features and simple SVM classifier. In order to compare with the previous results, I used the same leave-one-out cross validation protocol. This protocol has a serious problem. In the UvA-NEMO dataset, one parent has multiple children. It is unavoidable that when taking out a pair of relatives for verification, the video pair of the same parent and another child is left in the training set, causing serious leakage of the test set and resulting in unreliable results. Instead, leave-one-family out should be used. As shown in Section 3.1.3, the protocol we used in the subsequent experiments avoids the above problem.

Nonetheless, our positive experimental data hint at the potential of this solution, which we believe will greatly improve the performance of this approach when large datasets are available.

For future work, we suggest we first test our expression matching approach in a larger dataset with different facial expressions and collected in various conditions. Then try different hard negative sample mining methods while training as we have enough data. Another optional strategy is to abandon the Siamese structure and train multiple networks with the same structure but not sharing weights to deal with the problem of differences between parents and children due to gender and age in some kin-types. This process can also be automated by adding age estimation to the model.

6. Conclusions

Kinship verification is a difficult but promising research problem. This thesis attempts to use advanced vision transformer models to solve the problem of kinship verification in smiling videos. Our results show that that temporal features in videos have a positive effect on kinship verification. We also demonstrate the better ability of the ViT model in extracting spatio-temporal features, and its similar ability to convolutional methods in extracting spatial features. We also proposed various approaches including self-supervised pre-training, using matched expression sequences for training and different loss functions during training. However, the accuracy of our model is limited by the very limited size of this dataset, but we believe that larger datasets will largely improve this situation.

Bibliography

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846.

Baltrusaitis, T., Zadeh, A., Lim, Y. C., and Morency, L.-P. (2018). Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 59–66. IEEE.

Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4.

Borges, P. V. K., Conci, N., and Cavallaro, A. (2013). Video-based human behavior understanding: A survey. *IEEE transactions on circuits and systems for video technology*, 23(11):1993–2008.

Boutellaa, E., López, M. B., Ait-Aoudia, S., Feng, X., and Hadid, A. (2017). Kinship verification from videos using spatio-temporal texture features and deep learning. *arXiv preprint arXiv:1708.04069*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cao, C., Weng, Y., Lin, S., and Zhou, K. (2013). 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):1–10.

Dehghan, A., Ortiz, E. G., Villegas, R., and Shah, M. (2014). Who do i look like? determining parent-offspring resemblance via gated autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1757–1764.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dibeklioglu, H. (2017). Visual transformation aided contrastive learning for video-based kinship verification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2459–2468.

Dibeklioglu, H., Ali Salah, A., and Gevers, T. (2013). Like father, like son: Facial expression dynamics for kinship verification. In *Proceedings of the IEEE international conference on computer vision*, pages 1497–1504.

Dibeklioğlu, H., Salah, A. A., and Gevers, T. (2012). Are you really smiling at me? spontaneous versus posed enjoyment smiles. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision – ECCV 2012*, pages 525–538, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Ertugrul, I. Ö. and Dibeklioglu, H. (2017). What will your future child look like? modeling and synthesis of hereditary patterns of facial dynamics. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*, pages 33–40. IEEE.

Fang, R., Tang, K. D., Snavely, N., and Chen, T. (2010). Towards computational models of kinship verification. In *2010 IEEE International conference on image processing*, pages 1577–1580. IEEE.

Freitas Pereira, T. d., Anjos, A., Martino, J. M. D., and Marcel, S. (2012). Lbp- top based countermeasure against face spoofing attacks. In *Asian Conference on Computer Vision*, pages 121–132. Springer.

Gan, Y., Luo, Y., Yu, X., Zhang, B., and Yang, Y. (2021). Vidface: A full-transformer solver for video facehallucination with unaligned tiny snapshots. *arXiv preprint arXiv:2105.14954*.

Gao, X., Su, Y., Li, X., and Tao, D. (2010). A review of active appearance models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(2):145–158.

Girdhar, R., Carreira, J., Doersch, C., and Zisserman, A. (2019). Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253.

Goyal, A. and Meenpal, T. (2018). Detection of facial parts in kinship verification based on edge information. In *2018 conference on information and communication technology (CICT)*, pages 1–6. IEEE.

Goyal, A. and Meenpal, T. (2020). Patch-based dual-tree complex wavelet transform for kinship recognition. *IEEE Transactions on Image Processing*, 30:191–206.

Guo, G. and Wang, X. (2012). Kinship measurement on salient facial features. *IEEE Transactions on Instrumentation and Measurement*, 61(8):2322–2325.

Hansen, F., DeBruine, L. M., Holzleitner, I. J., Lee, A. J., O'Shea, K. J., and Fasolt, V. (2020). Kin recognition and perceived facial similarity. *Journal of Vision*, 20(6):18–18.

Jacob, G. M. and Stenger, B. (2021). Facial action unit detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7680–7689.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. (2021). Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*.

Kohli, N., Singh, R., and Vatsa, M. (2012). Self-similarity representation of weber faces for kinship classification. In *2012 IEEE fifth international conference on biometrics: theory, applications and systems (BTAS)*, pages 245–250. IEEE.

Kohli, N., Yadav, D., Vatsa, M., Singh, R., and Noore, A. (2018). Supervised mixed norm autoencoder for kinship verification in unconstrained videos. *IEEE Transactions on Image Processing*, 28(3):1329–1341.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Li, H., Sui, M., Zhao, F., Zha, Z., and Wu, F. (2021a). Mvt: mask vision transformer for facial expression recognition in the wild. *arXiv preprint arXiv:2106.04520*.

Li, L., Feng, X., Wu, X., Xia, Z., and Hadid, A. (2016). Kinship verification from faces via similarity metric based convolutional neural network. In *International conference on image analysis and recognition*, pages 539–548. Springer.

Li, W., Wang, S., Lu, J., Feng, J., and Zhou, J. (2021b). Meta-mining discriminative samples for kinship verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16135–16144.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., and Hu, H. (2022). Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211.

Lu, J., Hu, J., Zhou, X., Zhou, J., Castrillón-Santana, M., Lorenzo-Navarro, J., Kou, L., Shang, Y., Bottino, A., and Vieira, T. F. (2014). Kinship verification in the wild: The first kinship verification competition. In *IEEE international joint conference on biometrics*, pages 1–6. IEEE.

Lu, J., Zhou, X., Tan, Y.-P., Shang, Y., and Zhou, J. (2013). Neighborhood repulsed metric learning for kinship verification. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):331–345.

Ma, J., Jiang, X., Fan, A., Jiang, J., and Yan, J. (2021). Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129(1):23–79.

Moujahid, A. and Dornaika, F. (2019). A pyramid multi-level face descriptor: application to kinship verification. *Multimedia Tools and Applications*, 78(7):9335–9354.

Murphy-Chutorian, E. and Trivedi, M. M. (2008). Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):607–626.

Qin, X., Tan, X., and Chen, S. (2015). Tri-subject kinship verification: Understanding the core of a family. *IEEE Transactions on Multimedia*, 17(10):1855–1867.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. (2020). Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. (2021). Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128.

Robinson, J. P., Shao, M., Wu, Y., Liu, H., Gillis, T., and Fu, Y. (2018). Visual kinship recognition of families in the wild. *IEEE Transactions on pattern analysis and machine intelligence*, 40(11):2624–2637.

Shao, M., Xia, S., and Fu, Y. (2011). Genealogical face recognition based on ub kinface database. In *CVPR 2011 workshops*, pages 60–65. IEEE.

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27.

Suh, Y., Han, B., Kim, W., and Lee, K. M. (2019). Stochastic class-based hard example mining for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7251–7259.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

Sun, Y., Li, J., Wei, Y., and Yan, H. (2018). Video-based parent-child relationship prediction. In *2018 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE.

Tola, E., Lepetit, V., and Fua, P. (2009). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830.

Tong, Z., Song, Y., Wang, J., and Wang, L. (2022). Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, C. and Wang, Z. (2021). Progressive multi-scale vision transformer for facial action unit detection. *Frontiers in Neurorobotics*, 15.

Wang, M., Li, Z., Shu, X., Jingdong, and Tang, J. (2015). Deep kinship verification. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6.

Wang, S. and Yan, H. (2020). Discriminative sampling via deep reinforcement learning for kinship verification. *Pattern Recognition Letters*, 138:38–43.

Wang, W., You, S., and Gevers, T. (2020). Kinship identification through joint learning using kinship verification ensembles. In *European conference on computer vision*, pages 613–628. Springer.

Wang, X. and Kambhamettu, C. (2014). Leveraging appearance and geometry for kinship verification. In *2014 IEEE international conference on image processing (ICIP)*, pages 5017–5021. IEEE.

Wei, Z., Xu, M., Geng, L., Liu, H., and Yin, H. (2019). Adversarial similarity metric learning for kinship verification. *IEEE Access*, 7:100029–100035.

Wu, X., Boutellaa, E., López, M. B., Feng, X., and Hadid, A. (2016). On the usefulness of color for kinship verification from face images. In *2016 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE.

Wu, X., Feng, X., Cao, X., Xu, X., Hu, D., López, M. B., and Liu, L. (2022). Facial kinship verification: A comprehensive review and outlook. *International Journal of Computer Vision*, pages 1–32.

Wu, Y. and Ji, Q. (2019). Facial landmark detection: A literature survey. *International Journal of Computer Vision*, 127(2):115–142.

Xia, S., Shao, M., and Fu, Y. (2011). Kinship verification through transfer learning. In *Twenty-second international joint conference on artificial intelligence*.

Xia, S., Shao, M., and Fu, Y. (2012a). Toward kinship verification using visual attributes. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 549–552. IEEE.

Xia, S., Shao, M., Luo, J., and Fu, Y. (2012b). Understanding kin relationships in a photo. *IEEE Transactions on Multimedia*, 14(4):1046–1056.

Xing, E., Jordan, M., Russell, S. J., and Ng, A. (2002). Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15.

Xue, F., Wang, Q., and Guo, G. (2021). Transfer: Learning relation-aware facial expression representations with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3601–3610.

Yan, H. and Wang, S. (2019). Learning part-aware attention networks for kinship verification. *Pattern Recognition Letters*, 128:169–175.

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503.

Zhang12, K., Huang, Y., Song, C., Wu, H., Wang, L., and Intelligence, S. M. (2015). Kinship verification with deep convolutional neural networks. British machine vision conference. BMVA Press.

Zhong, Y. and Deng, W. (2021). Face transformer for recognition. *arXiv preprint arXiv:2103.14803*.