UNIVERSITÄT ZU LÜBECK

From the Institute of Neuro- and Bioinformatics
of the University of Lübeck
Director: Prof. Dr. rer. nat. Thomas Martinetz

Novel Machine Learning Methods for Video Understanding and Medical Analysis

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck

from the Department of Computer Sciences and Engineering

Submitted by

Yaxin Hu
from Ningguo, Anhui, China

Lübeck, 2024

First referee:	Prof. Dr.-Ing. Erhardt Barth
Second referee:	Prof. Dr. phil. Mattias Heinrich
Chairperson:	Prof. Dr. rer. nat. Stefan Fischer
Date of oral examination:	25.06.2025
Approved for printing.	26.06.2025

DECLARATION

I here by declare that the thesis entitled “Novel Machine Learning Methods for Video Understanding and Medical Analysis” submitted by me, for the award of the degree of *Dr. rer. nat.* to the University of Lübeck is a record of bonafide work carried out by me under the supervision of Prof. Erhardt Barth, Institute for Neuro- and Bioinformatics, University of Lübeck.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Lübeck

Date: 15/10/2024

Signature of the Candidate

(Yaxin Hu)

CERTIFICATE

This is to certify that the thesis entitled "Novel Machine Learning Methods for Video Understanding and Medical Analysis" submitted by Ms.Yaxin Hu in Institute for Neuro- and Bioinformatics, University of Lübeck, Lübeck, Germany for the award of the degree of *Dr. rer. nat.*, is a record of research work carried out by her under my supervision, as per the University of Lübeck code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: Lübeck

Date: 20/11/2024

Signature of the Supervisor

(Prof. Dr. Erhardt Barth)

ZUSAMMENFASSUNG

Künstliche Intelligenz hat sich im letzten Jahrzehnt rasant weiterentwickelt und ist in viele Aspekte des Lebens eingedrungen, vor allem in Bereiche wie Mensch-Computer-Interaktion, virtuelle Realität, autonomes Fahren und intelligente medizinische Systeme. Bei Videos handelt es sich um hochdimensionale Daten, die eine Dimension mehr haben als Bilder und daher mehr Rechenressourcen erfordern. Da immer mehr hochwertige, groß angelegte Videodatensätze veröffentlicht werden, hat sich die Analyse von Videos zu einer aktuellen Forschungsrichtung entwickelt. Es gibt inzwischen viele erfolgreiche Ansätze zur Erkennung von dynamischen Inhalten in Videos.

Bei unserer Arbeit konzentrieren wir uns darauf, neue Ansätze und Architekturen für die Analyse von Videos vorzuschlagen und deren Anwendungen in der Medizin zu untersuchen. Wir führen eine neue RGB_t-Abtaststrategie ein, um mehr zeitliche Informationen in einzelnen Bildern zu integrieren, ohne die Rechenlast zu erhöhen und untersuchen verschiedene Farbabtaststrategie, um die Erkennungsleistung weiter zu verbessern. Wir finden, dass Einzelbilder mit zeitlichen Informationen, die durch Fusion der grünen Kanäle aus zeitlich verschiedenen Einzelbildern gewonnen werden, die besten Ergebnisse erzielen. Wir verwenden Bereiche unterschiedlicher Größe, um zeitliche Information besser einzubetten, ohne die Rechenleistung zu erhöhen. Wir führen außerdem ein neues aus der Hirnforschung inspiriertes Neuronenmodell ein. Wir haben insgesamt eine neue räumlich-zeitliche Netzwerk-Architektur vorgeschlagen, die es 2D-CNNs ermöglicht, zeitliche Informationen zu nutzen. Alle genannten Methoden werden anhand von mindestens zwei Benchmark-Datensätzen evaluiert und weisen alle eine verbesserte Leistung auf.

Wir konzentrieren uns auch auf die Anwendung unserer Netzwerke in der Medizin. Wir verwenden unsere Netzwerk-Architektur welche örtlich-zeitliche Schnitte durch das Video verwendet für die Analyse von Glaukomen und Sehbehinderungen und wir stellen fest, dass Sehbehinderungen das Gehverhalten von Menschen beeinflussen können und somit das Gehverhalten diagnostisch relevant wird. Wir entwerfen außerdem ein KI-Modell zur Diagnose von Psychosen und zeigen, dass es möglich ist vorherzusagen, ob Risikopatienten tatsächlich eine Psychose entwickeln.

ABSTRACT

Artificial intelligence has developed rapidly over the past decade and has penetrated into nearly every aspect of life. New applications in areas such as human-computer interaction, virtual reality, autonomous driving and intelligent medical systems have emerged in large numbers. Video is a kind of high-dimensional data, which has one more dimension than images, requiring more computing resources. As more and more high-quality large-scale video datasets are released, video understanding has become a cutting-edge research direction in the computer vision community. Action recognition is one of the most important tasks in video understanding. There are many successful network architectures for video action recognition.

In our work, we focus on proposing new designs and architectures for video understanding and investigating their applications in medicine. We introduce a novel RGB_t sampling strategy to fuse temporal information into single frames without increasing the computational load and explore different color sampling strategies to further improve network performance. We find that frames with temporal information obtained by fusing the green channels from different frames achieve the best results. We use tubes of different sizes to embed richer temporal information into tokens without increasing the computational load. We also introduce a novel bio-inspired neuron model, the Min-Block, to make the network more information selective. Furthermore, we propose a spatiotemporal architecture that slices videos in space-time and thus enables 2D-CNNs to directly extract temporal information. All the above methods are evaluated on at least two benchmark datasets and all perform better than the baselines.

We also focus on applying our networks in medicine. We use our slicing 2D-CNN architecture for glaucoma and visual impairments analysis. And we find that visual impairments may affect walking patterns of humans thus making the video analysis relevant for diagnosis. We also design a machine learning model to diagnose psychosis and show that it is possible to predict whether clinical high-risk patients would actually develop a psychosis.

Keywords: *Deep Learning*, *Transformers*, *CNNs*, *Video Understanding*, *MRI Images*.

ACKNOWLEDGEMENT

First, I would like to express my sincere thanks to my supervisor Prof. Erhardt Barth, without his patient guidance and continuous encouragement, my work would not successfully completed. He has helped me a lot with my life in a new country and he has revised my papers in detail every time.

I would like to thank all colleagues at the Pattern Recognition Company GmbH and all the colleagues at the Institute for Neuro- and Bioinformatics, for motivating me to carry out and complete my research, and also for providing me with knowledge, computational resources and many other resources needed for my research.

I would also like to thank all my colleagues from OptiVisT EU, for their friendship, support and encouragement. I like to acknowledge the support given by Prof. Frans W. Cornelissen, Prof. Michael Hoffmann and my colleagues Ahmet Burak Kurt and Safa Andac throughout the Secondments. And I want to thank Andrea, Anna, Marcin and Ola for supporting me whenever I needed.

I would like to thank my parents for encouraging me all the time and supporting me in doing what I want to do, as well as for their patience and understanding. I would also like to thank all my friends for their constant encouragement and support along with friendship and kindness.

This project received funding from European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No.955590. Without this funding, this research would be impossible completed. Thanks to it, I have attended a lot of interesting workshops and had collaborations with many scientists in other field from different institutes and countries.

'Always remember that the answer must be in the attempt.'

Place: Lübeck

Date: 15/10/2024 **Yaxin Hu**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

In the past decade, with the rise of the self-media industry and the increase in storage capacity, a large number of high-quality and large-scale video datasets have emerged. With the increase in computing power, researchers are increasingly interested in research on video understanding tasks, and many great machine learning algorithms and deep learning networks have been proposed. Video is a kind of high-dimensional data because it has an additional time dimension, which greatly increases the difficulty of video processing and analysis. Similarly, medical imagery such as MRI data is also high dimensional. Usually, AI breakthroughs in a field can be used in other fields, which illustrates that methods performing well in the field of video understanding can also be transferred to medical high-dimensional image analysis. Human action recognition is an important task in video understanding and has been an active research field for many years. Therefore, we here focus on human action recognition tasks.

The aim of this task is to analyze the ongoing actions performed in videos. The actions consist of movements, gestures, interactions, and activities conducted usually by humans (Aggarwal and Ryoo 2011). Movements usually refer to physical activities performed by a person, such as walking, jumping, running, etc. Movement recognition is commonly used in sports analytics and intelligent security systems. Gestures usually refer to human hand gestures such as clicking, pointing, and throwing performed by fingers, palms, and arms. But in a broad sense, gestures also include the movements made by the head, legs and other human body parts. Gesture recognition enables users to interact with devices without physical contact or complex input devices, thereby it is widely used in virtual reality and human-computer interaction. Interaction refers to actions between people or between people and devices, including facial expressions, body language, etc. The difficulty of interaction recognition is not only to recognize individual actions but also to understand situational relationships. Activities are mixtures of gestures, movements or interactions, such as cooking, doing housework, etc. Activity recognition involves the recognition and understanding of complex activities and is used in smart home and health monitoring system.

Human actions are diverse and often complex, making it difficult to accurately recognize and understand actions in videos. These actions have both strong intra- and

inter-class variations. The same action can be performed by different people with different postures at various speeds in different scenarios. Moreover, the same action appears differently from different shooting angles and distances. These videos may be filmed from above, from the side, or from the front of the participants. The action looks very different due to the different viewpoints. Some different actions may have some similar movement patterns or be performed in similar scenarios, these similarities make them difficult to be distinguished. In addition, factors such as background noise, camera movement, lighting changes, and occlusions can also affect the performance of action recognition tasks.

For video action recognition tasks, temporal reasoning is very important. For example, the actions 'pushing' and 'pulling', and the actions 'opening the door' and 'closing the door' look similar on still frames. However, by analyzing the context in the video, it can be inferred that there are two actions. Complete action execution in video provides more reliable and detailed information than frozen actions on static images. Therefore, videos are more useful than images when it comes to recognizing and understanding human actions. Therefore, extracting spatial features from a single frame is not sufficient to represent the video, it is also necessary to capture the changes among frames, that is, the information in the temporal dimension. Human actions usually involve long-term space-time interactions. Thus, utilizing spatiotemporal information is crucial for human action recognition.

Video action recognition tasks involve extracting spatiotemporal representations of videos and making classification decisions. Spatiotemporal representations of context is obtained by extracting effective features from video. The methods of obtaining spatiotemporal representation greatly affects the performance of action recognition and computational efficiency of the entire model. Therefore, we need networks that can extract valid spatiotemporal features and make precise decisions by using these features.

Despite similarities to still image processing, video understanding is much more complex. Since videos have one more dimension than images, the networks for videos processing require huge computation resources and longer training time. Another challenge is that the videos vary in length, but current networks can only process inputs of the same length. Effective sampling to reduce video redundancy and computational cost while retaining frames containing useful information is still a research direction worth exploring. The balance between high accuracy and computational cost is another important topic for video action recognition tasks.

In order to apply the model to actual scenarios, the generalization ability and real-time performance of action recognition are crucial. Generalization performance refers to the ability of a trained model to make accurate decisions on unseen data and it describes model performance on data independent of the training data. Most of the currently popular models are trained on public large-scale labeled datasets. When they are

applied to real scenarios, they often cannot generalize well. Because real-world data comes from a wider range of sources, is more diverse, and has varying data quality. Real-time performance refers to the ability of models to process data and display results instantaneously without any delays or interruptions (Maier-Hein et al. 2013). The real-time performance is necessary for the model to be applied in the real world. The delayed response of the model brings bad experiences to users, such as virtual reality and human-computer interaction, and even causes dangerous situations, such as delayed decision-making of surgical robots.

Video action recognition is a very promising research field and understanding human behavior in visual data helps make progress in related research fields. These fields include video retrieval and recommendation, game and entertainment, medical care, education, etc. Smart Medicine is one of the most cutting-edge cross-cutting research directions. The video action recognition architecture we studied can be applied to surgical robots for surgical videos segmentation and workflow recognition, thereby alerting surgeons possible complications, reducing their operative mistakes and supporting decision making. Moreover, the video architectures can be used in MRI data for detecting tumors and analysing neurodegenerative diseases such as Alzheimer's disease, Parkinson's disease.

The rest of the thesis is structured as follows. In the Chapter 2, we review the classic machine learning methods and convolutional neural network architectures, and detail state-of-the-art video transformer architectures. In Chapter 3, we propose three novel designs to improve the video transformer architecture. In Chapter 4, we present a novel 2D-CNN architecture operating on salient spatiotemporal slices for Video Understanding. In Chapter 5, we apply the architecture proposed in Chapter 4 to glaucoma diagnosis and visual impairment detecting. In Chapter 6, we investigate the effective use of color and temporal information for videos and evaluate our ideas on 2D-, 3D-CNNs and Video Transformers. In Chapter 7, we design a machine learning model to diagnose psychosis and predict the transition of psychosis.

CHAPTER 2

Related Work

With the rapid increase of video resources, storage capacity and computing power, video analysis has become an important and inevitable task in the field of computer vision. In recent years, deep learning methods have made great achievements in natural language processing and image processing. Therefore, researchers have also changed their research focus from initially using handcrafted features and machine learning classifiers for video analysis to different designs that apply and extend 2D neural networks to the time domain.

In this chapter, we first briefly introduce the hand-crafted feature based methods. We then describe the neural network architectures based on CNNs in detail and introduce the various classic CNN-based architectures for video understanding tasks. Next, we explain state-of-the-art networks for video processing - Video Transformers. In the end, we summary the most common video benchmark datasets for video understanding.

2.1 Hand-crafted Features

Traditional machine learning methods for action recognition tasks typically combine handcrafted features with machine learning classifiers (Zhu et al. 2020). There are two main methods for extracting handcrafted features. One is based on holistic features, and the other is based on local features.

2.1.1 Holistic Features

The action of subject in a video not only contains spatial information at different time, but also dynamic information. Extracting holistic features requires precise localizing and tracking to capture the motion information of the entire human body, usually based on computing silhouette, shape and optical flow. Bobick and Davis (2001) proposed temporal templates that combine motion energy images (MEI), which records the presence of motion at each pixel and motion history images (MHI), which shows the motion location and path as it progresses based on MEI. Yilmaz and Shah (2005) presented spatiotemporal action volumes (STV) by projecting the 3D boundary as 2D contour in image plane and obtained a set of action descriptors based on the sign of Gaussian

and mean curvature by analyzing the differential geometry of local volume surfaces. Optical flow is the apparent motion pattern of image objects between two consecutive frames caused by object or camera motion (Warren and Strelow 2013) and is a way to characterize and quantify the motion in videos.

2.1.2 Local Features

However, holistic features based methods are sensitive to noise, perspective changes, and occlusion. Local features can alleviate these problems by directly extracting features from local regions that are more informative and salient. Laptev (2005) extended spatiotemporal interest points (STIP) to the spatiotemporal domain to capture local feature and introduced a Laplacian operator for scale selection in space-time. Dollár et al. (2005) extracted a cuboid at each point of interest as a descriptor to contain all the information needed to represent the corresponding action. Scovanner et al. (2007) introduced a 3D SIFT descriptor to better represent the 3D nature of video data and used a bag of words paradigm to improve model performance. Klaser et al. (2008) extended Histogram of Oriented Gradient (HOG) features to 3D local descriptors based on histograms of oriented three-dimensional gradients in space-time. Wang and Schmid (2013) proposed using speeded-up robust features (SURF) descriptors and dense optical flow to match feature points between frames to estimate camera motion and perform calibration to improve the performance of descriptors such as Histogram of Optical Flow (HOF).

2.2 Convolutional Neural Networks

Compared with hand-crafted features, deep learning networks can automatically extract features and are more robust and more suitable for large datasets and complex scenarios. With the breakthrough of convolutional neural networks (CNNs) for image processing, researchers have adapted them to video processing. Videos have an additional temporal dimension compared to images, which is a key issue in applying CNNs to videos. There are several research directions in extending CNNs to the time domain for video analysis.

2.2.1 Two-Stream Architectures

One research direction is to use two-stream networks, typically one stream is used to capture spatial information and the other stream is used to model temporal dependencies and then a late fusion is applied to the two streams to obtain the final spatiotemporal representation of the video. The typical architecture of two-stream networks is shown in Figure 2.1. DeepVideo is one of the earliest attempt, Karpathy et al. (2014) proposed DeepVideo that investigated four different approaches to fuse temporal informa-

tion and a multiresolution architecture, which consisted of a context stream processing low-resolution image and a fovea stream operating high-resolution center crop.

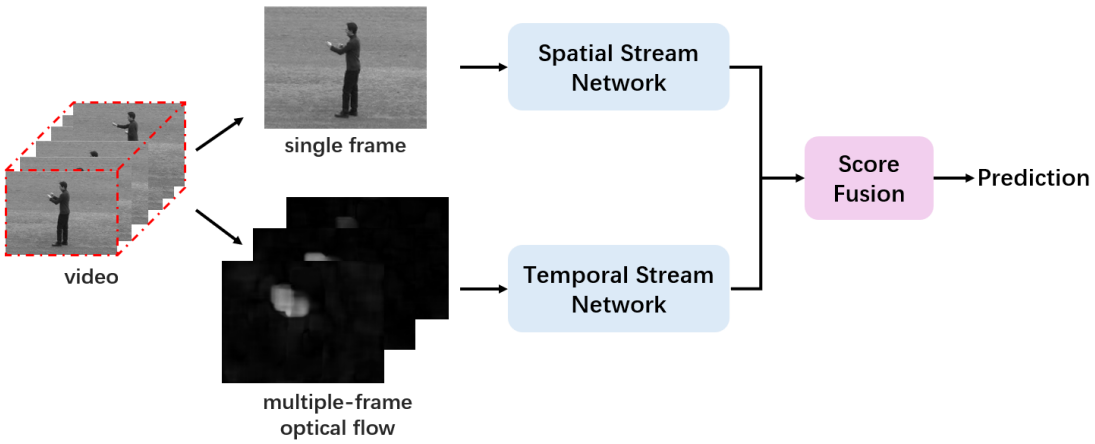


Fig. 2.1 Architecture of a two-stream network.

Optical flow is used as the input for the temporal stream in two-stream networks because it is able to represent the motion information of moving objects and is not affected by background and camera motion. Simonyan and Zisserman (2014) explored a two-stream network with a spatial stream taking a single still frame as input to extract spatial information and a temporal stream using optical flow (Horn and Schunck 1981) as input to capture dynamics between video frames. However, optical flow is computationally expensive and time consuming, making it is unsuitable for large dataset training and end-to-end learning.

2.2.2 Segment-based Architectures

Two-stream networks have shown that CNNs can be applied to video processing. However, they have a limited ability to extract temporal information. Therefore, a novel sampling strategy was proposed to first uniformly segment the video into multiple snippets, and then randomly select one frame from each snippet to form a clip that represents the video. The clips obtained through this strategy not only cover the entire temporal dimension of the video but also make the network more robust.

Wang et al. (2016*a*) presented the Temporal Segment Network (TSN) that used a spatial ConvNet to capture spatial features from sparsely sampled frames, a temporal ConvNet to model temporal dependencies from not only stacked optical flow fields but also RGB differences. Compared to previous studies, they designed a sparse frame sampling strategy and used a consensus aggregation module to model longer time series. Figure 2.2 shows the architecture of the TSN. Lin et al. (2019) introduced a Temporal Shift Module (TSM) that used TSN as backbone and shifted feature map along temporal dimension to better model context. Liu et al. (2021) designed a Temporal Adaptive

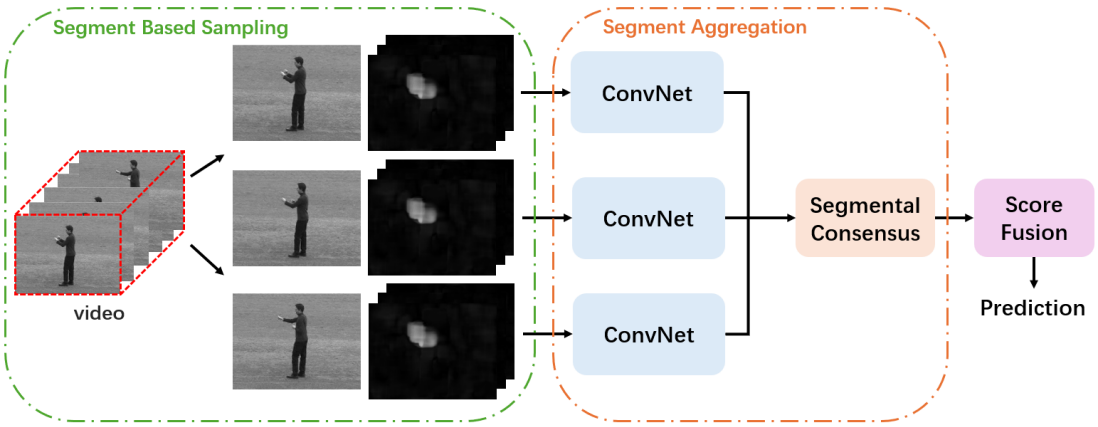


Fig. 2.2 Architecture of TSN.

Module (TAM) with two branches, a local branch to model short-term dynamics for specific video operations, which is location sensitive, and a global branch to generate an adaptive kernel by using long-range dynamics, which is location invariant.

2.2.3 CNN-RNN Architectures

Recurrent neural networks (RNNs) contain recurrent structure that allows information to be passed among neurons, enabling RNNs to effectively process time series data and preserve the contextual information of the data. Combinations of CNNs and RNNs are often used in video analysis, especially CNNs combined with long short-term memory (LSTM) networks (Hochreiter and Schmidhuber 1997). The typical architecture of a CNN-RNN network is shown in Figure 2.3. LSTM contains memory units and gate functions that enable it to model longer temporal information. A CNN architecture first takes frames as input to extract spatial representation and then a LSTM network captures temporal representation based on the output of the CNN, and at last a MLP classifier is applied for prediction (Donahue et al. 2015, Yue-Hei Ng et al. 2015). However, the temporal memory of RNNs is quite short and cannot capture long-range temporal information. Furthermore, RNNs process spatial and temporal information serially rather than in parallel, which is both computationally expensive and time-consuming.

2.2.4 Other 2D-CNN Architectures

There are also studies that explore other methods to model dynamics. Temporal downsampling can reduce video redundancy to a certain extent but also lead to the loss of temporal details. Andrearczyk and Whelan (2018) proposed a slicing approach to directly capture dynamics from slices obtained in three orthogonal planes by using unsupervised learning (PCA) and a shallow CNN. Figure 2.4 shows the architecture of this network. By slicing the video from different perspectives, CNNs have the capacity to

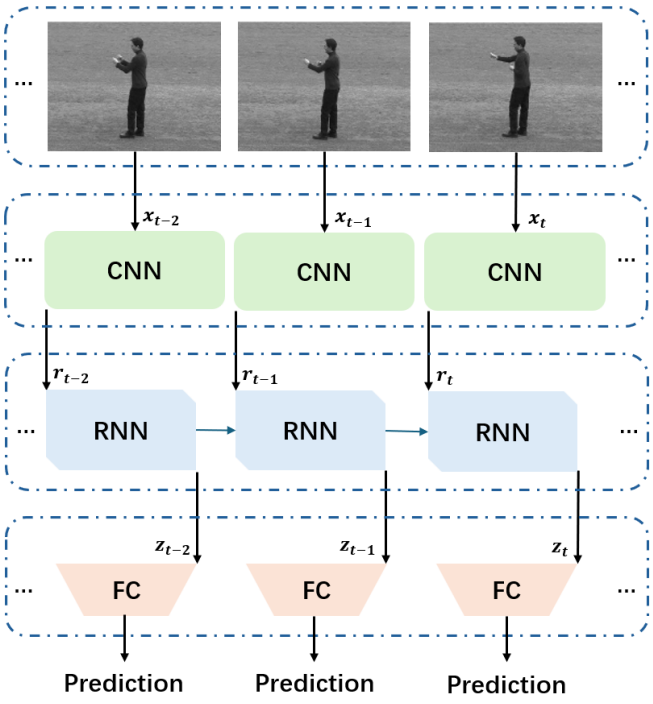


Fig. 2.3 Architecture of a CNN-RNN network.

directly capture temporal feature on simple frames.

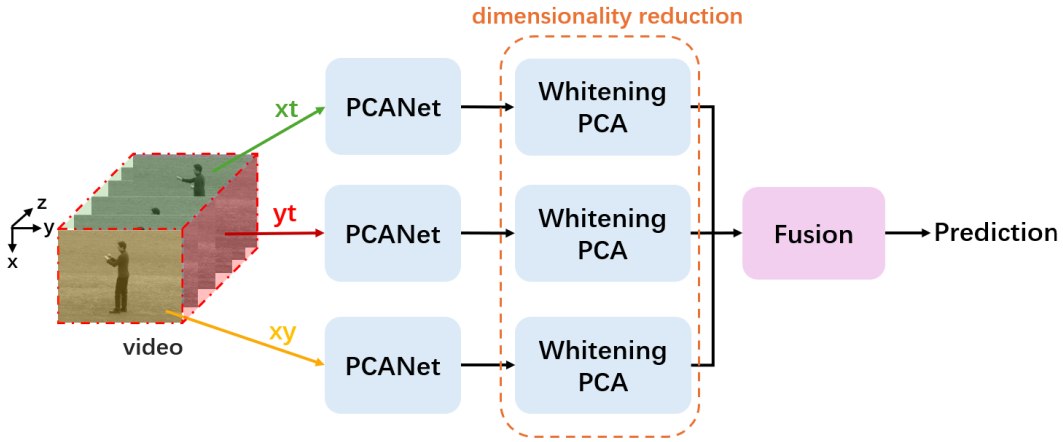


Fig. 2.4 Architecture of the PCANet.

2.2.5 3D-CNN Architectures

In addition to the above use of 2D-CNN architectures to process videos, a more straightforward method is to extend the 2D kernels to 3D kernels, so that the 3D-CNN (Yao et al. 2015) can capture spatial and temporal information simultaneously. Tran et al. (2015) explored a 10-layer 3D-CNN architecture, trained it on the large dataset Sport1M (Karpathy et al. 2014) and used the learned feature extractors for various video tasks.

C3D generalized well on other tasks, but the training on the Sport1M dataset is very time-consuming and computationally expensive. The benefit of expanding the kernel is that 3D-CNNs can directly use 2D-CNNs' weights pretrained on large image datasets by simply inflating them. Carreira and Zisserman (2017) presented a Two-Stream Inflated ConvNet (I3D) to learn seamless spatio-temporal features from videos using a ImageNet-pretrained Inception-V1 (Ioffe 2015) as backbone. They post-pretrained the network on kinetics datasets (Kay et al. 2017) and evaluated I3D on smaller datasets, such as UCF101 (Soomro et al. 2012*a*) and HMDB51 (Kuehne et al. 2011). And they also compared the results of taking only RGB or only optical flow or both of them as input and concluded that the two-stream architecture achieved the best results. The architecture of I3D is shown in Figure 2.5. I3D is a milestone in the development of video processing, after large-scale datasets have become the benchmarks for video understanding tasks.

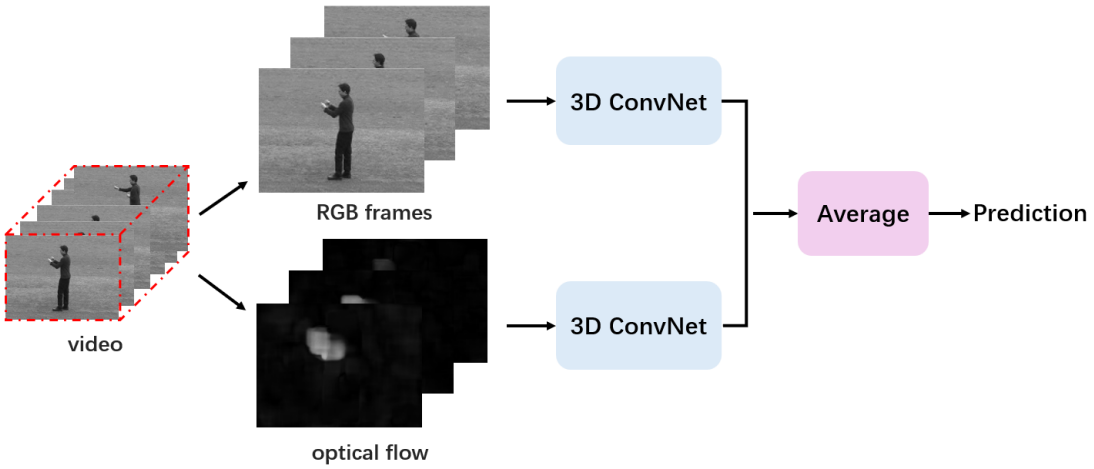


Fig. 2.5 Architecture of I3D network.

After I3D, a lot of 3D-CNN variants have been proposed. ResNet3D (Hara et al. 2018) directly inflated a 2D-ResNet into 3D and also adopted the weights of 2D-ResNet pretrained on ImageNet. Feichtenhofer (2020) proposed a simple step-wise expansion approach for 2D-CNNs along space, time, width, and depth to obtain a good trade-off between accuracy and complexity.

Inspired by the primate visual system, Feichtenhofer et al. (2019) argued that different temporal speeds should be taken into account. Thus, they proposed a two-pathway SlowFast Network, a slow pathway operating at low frame rate to capture spatial semantics, and a fast pathway operating at high frame rate to capture motion at fine temporal resolution. Figure 2.6 shows the architecture of the SlowFast network. Although the SlowFast network has two pathways, it is not a two-stream network. It is essentially a single-stream network operating at two different framerates.

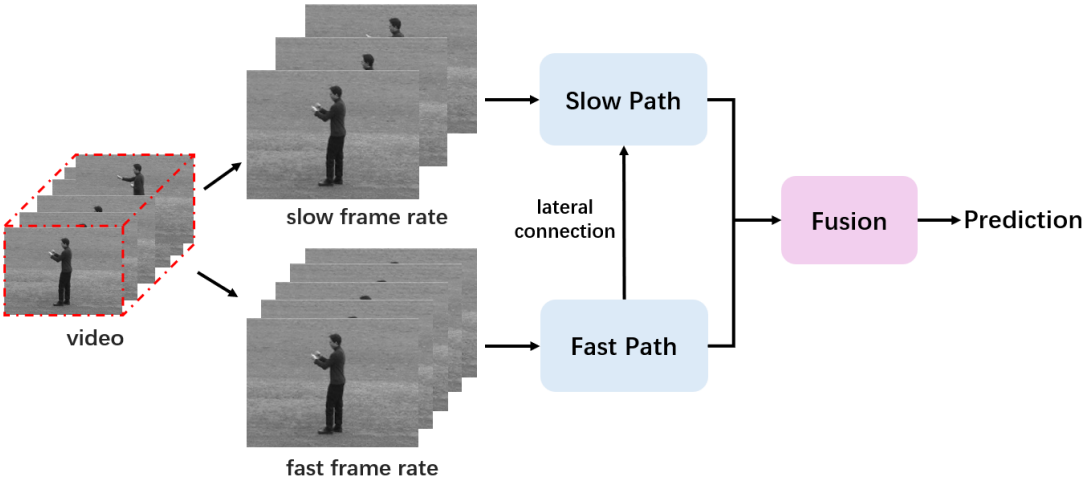


Fig. 2.6 Architecture of the SlowFast network.

2.2.6 3D Convolution Factorization

3D-CNNs are diffcult to train and require high computational costs. Some ideas about 3D factorization were explored. One idea is to factorize a 3D spatiotemporal kernel into a 2D spatial kernel and a 1D temporal kernel. The factorization of space and time is shown as Figure 2.7. Tran et al. (2018) presented an R(2+1)D network to sequentially extract spatial features and capture temporal features, and introduced more non-linearity by using additional ReLUs to enable the network to learn more complex semantics. The results showed that the R(2+1)D network performed better compared to networks of similar capacity, and this advantage became more obvious as the network deepens.

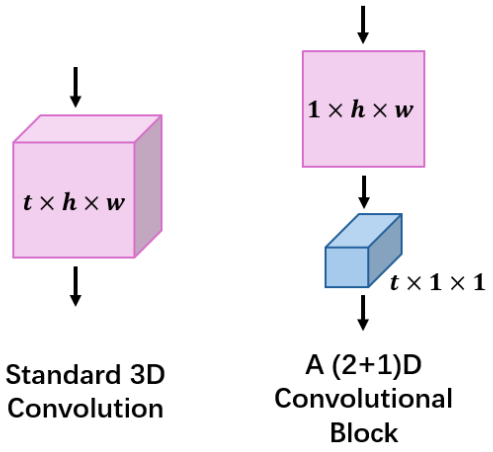


Fig. 2.7 Factorization of space and time.

Another idea is to split a full convolution layer into two separate layers, a depth-wise convolution layer that applies a single convolutional filter per input channel, and a pointwise convolution layer that applies 1×1 convolution. Figure 2.8 shows the fac-

torization of space-time and channel. Sandler et al. (2018) proposed MobileNetV2 that explored depthwise separable convolutions, linear bottlenecks together with inverted residuals blocks to obtain a good balance of performance and efficiency. Tran et al. (2019) designed a ChannelSeparated Convolutional Network (CSN) that explored how the channel interactions affected the performance and found that factorizing convolution improved the accuracy and reduced the computational loads.

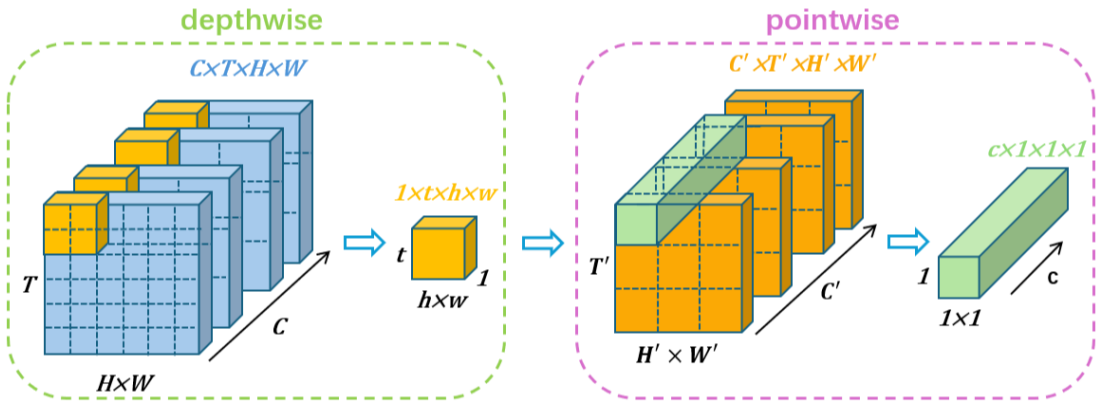


Fig. 2.8 Factorization of space-time and channel.

Unfortunately, 3D-CNNs have limitations in modeling long-term temporal dependencies due to its limited receptive field. Moreover, a 3D-CNN is computationally intensive and data hungry. Thus, new architectures are needed for modeling longer-term temporal information.

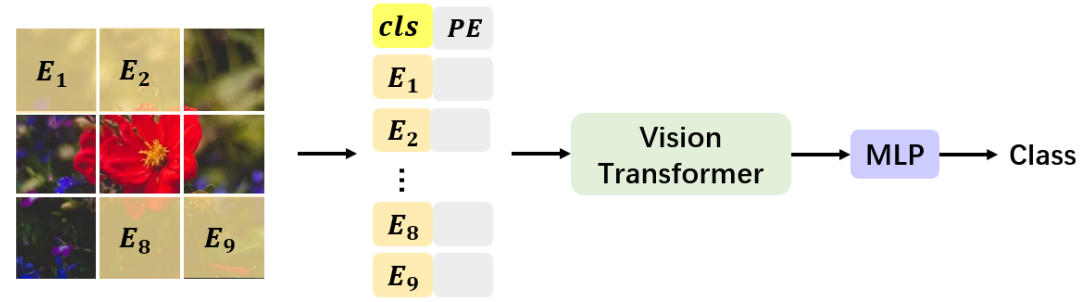
2.3 Video Transformers

The great success of Transformers in the field of natural language processing showed its excellent capacity in processing sequence data. This has also contributed to the development of Transformers in the field of computer vision. Soon, Transformers and its variants replaced the dominance of convolutional neural networks in the field of image processing, becoming the cutting-edge architectures for large image benchmarks such as ImageNet (Deng et al. 2009). Based on the achievements of Transformers in image processing and their ability to model long-range dependency, it is also considered a promising architecture for processing video. Naturally, Transformers have been also adapted to video understanding tasks and have quickly become the state-of-the-art architectures for almost all video benchmarks.

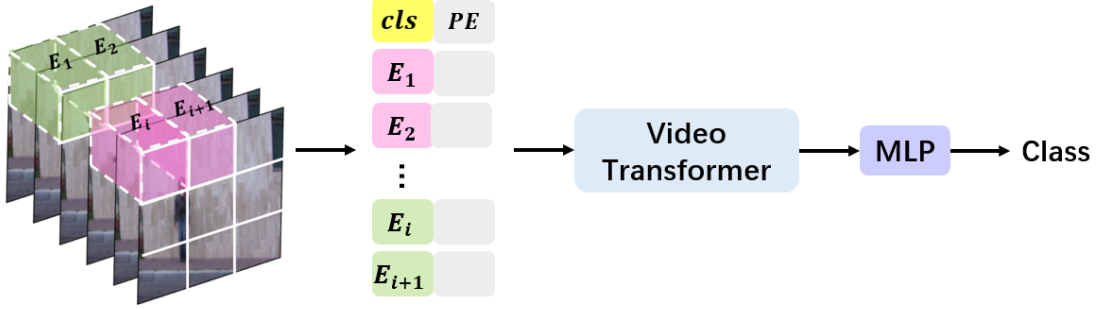
2.3.1 Tokenlization

Similar to natural language processing (NLP), input words or characters are represented as a sequence of tokens (Vaswani et al. 2023). Figure 2.9 shows the input tokenization

procedure for images and videos. For image processing tasks, input images are split into non-overlapping, fixed-size patches and these patches are then linearly embedded into tokens (Dosovitskiy et al. 2020*a*); for video understanding tasks, videos are represented as a series of sampled frames and divided into non-overlapping, fixed-size tubelets (it consists of patches with different time index in the same space, and the number of patches is greater than or equal to one.), and then these tubelets are linearly embedded as tokens (Arnab et al. 2021).



(a) Image tokenization.



(b) Video tokenization.

Fig. 2.9 Input tokenization for Vision Transformers (***cls*** represents class token, E_i represents the i^{th} token embedded from i^{th} patch or tubelet, and ***PE*** represents Position Embedding).

As shown in Figure 2.9 (b), the detailed description of video tokenization for an input of dimension $T \times H \times W$ is as following:

$$\mathbf{z} = [z_{cls}, \mathbf{E}x_1, \mathbf{E}x_2, ..., \mathbf{E}x_N] + \mathbf{PE}, \tag{2.1}$$

where $x_i \in \mathbb{R}^{h \times w}$ is the i^{th} non-overlapping video tubelet. $\mathbf{E}$ represents the projection performed on tubelets in order to obtain tokens, and $\mathbf{E}$ is usually performed by 3D convolution with kernel size $t \times h \times w$ and stride (t, h, w). $\mathbf{z}$ is the sequence of tokens obtained by projection and z_{cls} is the learned classification token. $\mathbf{PE} \in \mathbb{R}^{Num \times d}$ represents the learned position embedding, and $Num = \lfloor \frac{T}{t} \rfloor \times \lfloor \frac{H}{h} \rfloor \times \lfloor \frac{W}{w} \rfloor$ is the number of tubelets.

2.3.2 Architecture

Although there are many variants of Video Transformers, most of them are based on the architecture shown in Figure 2.10. After tokenization, the tokens are fed into a Transformer encoder, which consists of N sequentially connected Transformer blocks. Each block n contains multi-headed self-attention (MSA) (Vaswani et al. 2023), layer normalisation (LN) (Ba 2016), and Multilayer Perceptron (MLP). The formulas are defined as below:

$$\mathbf{y}^n = MSA(\mathbf{LN}(\mathbf{z}^n)) + \mathbf{z}^n, \tag{2.2}$$

$$\mathbf{z}^{n+1} = MLP(\mathbf{LN}(\mathbf{y}^n)) + \mathbf{y}^n, \tag{2.3}$$

$$\mathbf{Class} = MLPHead({z_{cls}}^N), \tag{2.4}$$

where $\mathbf{z}^n$ represents the tokens after tokenization layer or the output of the n^th Transformer block. $\mathbf{y}^n$ is the output of multi-headed self-attention (MSA) module. $\mathbf{z}^{n+1}$ represents the output of the $n+1^{th}$ Transformer block. At last the learned classification token z_{cls} is passed through a MLP head for classification.

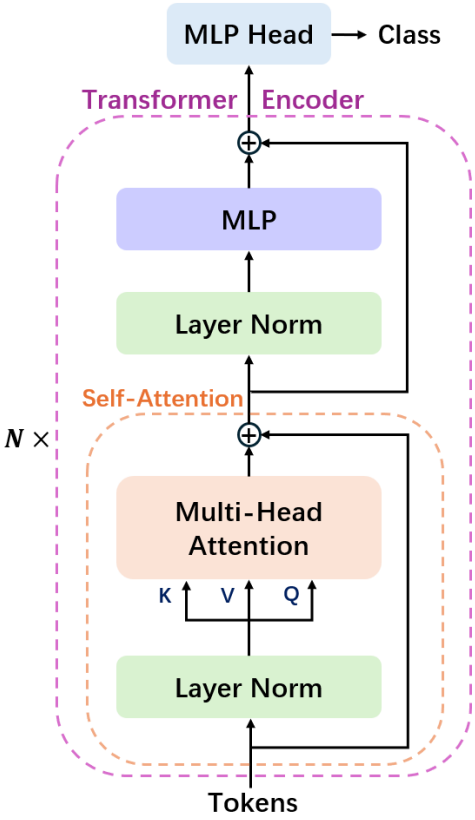


Fig. 2.10 Basic architecture of a Transformer encoder

Transformers succeed not only due to their architectures, but also because of their multi-headed self-attention (MSA) mechanism. Each head focuses on different parts

of the input, allowing the model to capture different information at the same time and learn more complex semantic representation. Suppose there are h heads, the dimension of each head is d and $\hat{\mathbf{z}}$ represents $\mathbf{z}$ after layer normalisation (LN).

$$MSA(\hat{\mathbf{z}}) = Concat(\mathbf{O}_1(\hat{\mathbf{z}}), \mathbf{O}_2(\hat{\mathbf{z}}), ..., \mathbf{O}_h(\hat{\mathbf{z}}))\mathbf{W}^{\mathbf{O}}, \tag{2.5}$$

$$\mathbf{O}_i = \mathbf{A}_i\mathbf{V}_i(\hat{\mathbf{z}}), \tag{2.6}$$

$$A_i = Softmax\left(\frac{\mathbf{Q}_i(\hat{\mathbf{z}})\mathbf{K}_i(\hat{\mathbf{z}})^{\mathsf{T}}}{\sqrt{d_k}}\right), \tag{2.7}$$

where O_i is the output of the i_{th} head in multi-headed self-attention (MSA). $Concat$ represents the concatenate operation of multiple heads. W^O is the weight matrix of the linear transformation.

2.3.3 Related Work

The capability of capturing long-term temporal dynamics has led to the rapid development of Transformers in the field of video processing. There are many video Transformer variants. Neimark et al. (2021) presented the VTN model for video recognition by adding a temporal attention-based encoder on top of a 2D spatial backbone to model temporal dependencies of extracted spatial features and use an MLP head for classification. VTN is similar to the CNN-RNN architecture, but uses an attention mechanism for temporal information instead of a RNN. Bertasius et al. (2021) proposed the convolution free architecture Timesformer, which is a fully self-attention-based video classification method. Following ViT (Dosovitskiy et al. 2020*a*), Timesformers obtain patches from frames and learn spatiotemporal features directly from a sequence of patches. Arnab et al. (2021) proposed the ViviT architecture that adapted the ViT (Dosovitskiy et al. 2020*a*) network pretrained on CLIP400 (Schuhmann et al. 2021) to videos; it introduced tubelet embedding on video clips to tokenize the spatiotemporal information simultaneously and explored four attention designs to find the most effective and efficient way of handling spatial attention and temporal attention.

In addition to the aforementioned early attempts to adapt Transformers to video, there are Transformer variants that introduce some novel video-specific designs. Fan et al. (2021) designed MViT, a channel-resolution scale model that progressively expands the channel capacity while reducing spatiotemporal resolution to extract low-level visual features at early layers and model complex semantic features at deep layers. Yan et al. (2022) presented the MTV architecture, which uses multiple encoders to extract spatiotemporal features from tokens obtained by tubelets of different dimensions, and uses cross-view fusion and a global encoder to fuse the features from different views. Li et al. (2022*a*) proposed UniFormer, which combines a Transformer with 3D convolution to capture local and global spatiotemporal features, allowing it to reduce

the redundancy of videos while modeling long-range dependencies. Piergiovanni et al. (2023) explored TubeViT that uses sparse video tubes of different sizes and a ViT encoder to work seamlessly with images and videos.

Local attention is also one of the research directions in Transformers. Liu et al. (2022) proposed Swin Transformers, using a 3D shifted windows based multi-head self-attention module to introduce cross-window connections for capturing the spatio-temporal locality of videos. There are also Transformers that focus on self-supervised learning to improve generalization performance. Tong et al. (2022) proposed VideoMAE, which works by first randomly masking most patches and then using an encoder operating on the visible patches and a light decoder processing all patches to reconstruct the input in the pixel space.

2.4 Multimodality

Due to the rapid increase in large networks and large-scale benchmark datasets of various modalities, multimodal models have become a promising research area. There are many different data modalities, including RGB, RGB-Depth, Optical Flow, audio, video, various signals and so on. The state-of-the-art architectures are currently defined by multi-modality Transformers. Wang et al. (2022) proposed video foundation model the InternVideo, which conducted self-supervised pretraining by using a video masked modeling module (Tong et al. 2022) and a video-language contrastive learning module, and then enhanced video representation by using supervised post-pretraining, and next used cross representation learning to unify two modules. Srivastava and Sharma (2024) presented OmniVec network that consists of a modality encoder to extract the features from modalities, a projection layer to project the features conditioned on meta tokens of modalities, a Transformer network to process the patches obtained from projection and a vectorizer to output embeddings for the original data point. Then the output can be used for different downstream tasks.

2.5 Benchmark Datasets

As computing power and storage capabilities increase, the scale and the quality of datasets grows. This further increases the complexity of deep learning networks. Figure 2.11 shows the most common datasets in video understanding tasks. Video datasets can be roughly divided into two categories.

One are scene-related datasets, such as UCF101 (Soomro et al. 2012*a*), HMDB51 (Kuehne et al. 2011) and Kinetics (Kay et al. 2017). The recognition of actions in scene-related datasets relies more on spatial information such as objects and backgrounds. For example, the model can predict the action 'playing basketball' correctly by recognizing

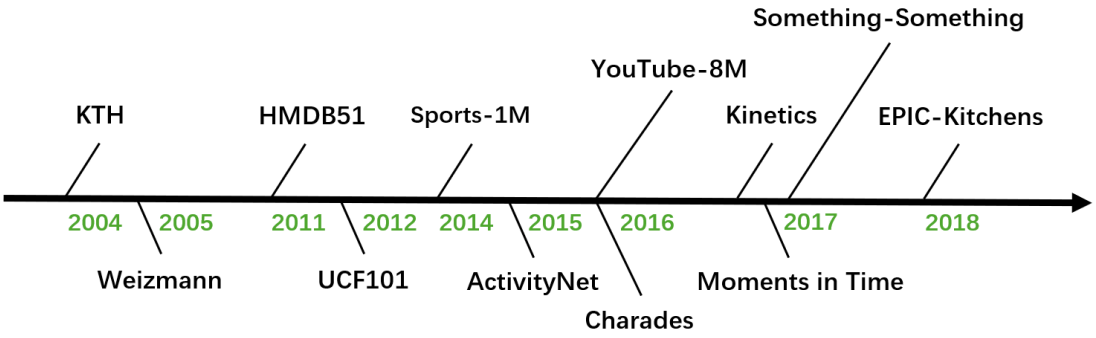


Fig. 2.11 An overview of representative datasets for action recognition.

the basketball, the human and other scenes related to basketball. Sometimes, the model can recognize the action by only using a still image of the video, without requiring much temporal information.

The other kind of dataset is temporal-related dataset, such as Something to Something V1/V2 (Goyal et al. 2017) and Moment in Time (Monfort et al. 2019). The actions in temporal-related datasets are more fine-grained. It is not easy for a model to predict the fine-grained actions correctly by only using spatial information. For example, if there is a single image with a table, a cup on the table and a human's hand on it. Even humans are not sure whether the action is "take a cup from the table" or "put a cup on the table." To predict action like this, the model needs to know the context of this single image. Thus, temporal information is needed for correct prediction.

The KTH (Schuldt et al. 2004) and Weizmann (Blank et al. 2005) datasets were commonly used for hand-crafted feature based models, both of them are very small and have controlled background. UCF101 (Soomro et al. 2012*a*) and HMDB (Kuehne et al. 2011) datasets were classic benchmarks for video action recognition tasks before I3D was proposed. Sports-1M (Karpathy et al. 2014) is one of the largest datasets commonly used to pretrain 3D-CNN architectures for evaluation on UCF101 and HMDB51 datasets. ActivityNet (Caba Heilbron et al. 2015) consists of untrimmed videos, each containing multiple activities. YouTube-8M (Abu-El-Haija et al. 2016) is currently the largest video dataset, its annotations are generated through retrieval methods. Charades (Sigurdsson et al. 2016) includes videos for daily indoor activities, and it is a multi-label dataset.

The other datasets in Figure 2.11 are most commonly used as benchmarks for Video Transformers. Kinetics Family (Kay et al. 2017) is the current mainstream benchmark for video analysis and usually used for post-pretraining for large networks. Something-Something (Goyal et al. 2017) is the most temporal dependent dataset, most of the actions in it require strong temporal reasoning. Moments in Time (Monfort et al. 2019) contains videos that involve not only people, but also animals, objects and natural phe-

nomena. EPIC-Kitchens (Damen et al. 2018) is a dataset of first-person vision of participants recorded by head-mounted cameras. These videos involve different kitchen tasks. Detailed information about the above datasets can be found in Appendix A.

CHAPTER 3

Novel Designs of Video Transformers for Action Recognition

3.1 Introduction

In action recognition tasks, actions can be very difficult for networks to classify. The same action can be performed by different subjects with different gestures at different speeds in different scenarios, meaning that network have to find the commonalities among these videos. And the different actions can also be performed by the same subject with similar motion patterns at similar speed in the same background, these similarities make the networks difficult to detect the subtle differences. Therefore, making good use of both spatial information and temporal information is crucial for video action recognition. A successful network must have the capacity to capture useful spatiotemporal features for correctly action classifications. Due to the additional temporal dimension of videos, much more computational resources are required by video processing networks. Thus, a good trade-off between accuracy and computational costs is a key issue for video action recognition tasks.

3D-CNNs can extract spatiotemporal features from a relatively small 3D neighborhood to capture local dependencies but have limitations in modeling global dependencies on video context due to the limited receptive field. Transformers can capture longer temporal information due to their self-attention mechanism, but they are limited in reducing local redundancies because all the input tokens are compared blindly.

The combination of 3D convolution and Transformer architecture can simultaneously capture local and global spatiotemporal features. UniFormerV2 (Li et al. 2022*a*) is a successful variant of Transformers combining 3D convolution and spatio-temporal self-attention to reduce the local redundancy and also capture long-time dependencies in videos. Thus, we choose UniFormerV2 as our backbone for further improving. However, like other Transformers, the performance of UniFormerV2 is also limited by the size of the dataset. Therefore, we aim to demonstrate the effectiveness of our three novel design ideas through relative improvements on small datasets.

The inputs of Video Transformers are usually video clips sampled from videos. The length of the clips is limited by the available computational resources but the performance is proportional to some extent to the length of the clips. Thus, we are motivated

to find a way to model longer temporal dependencies of videos without increasing much computational costs and to introduce a bio-inspired nonlinear connection between neurons that makes neurons more selective.

3.2 Methodology

Videos are essentially stacked images in the temporal dimension. They have additional temporal information, which makes the data higher dimensional and more redundant. Thus, video processing is more complicated and requires more computational resources than image processing. It is very important to get a good balance of network performance and computational costs.

In this section, we introduce three different novel design ideas that improve Video Transformer networks on action recognition tasks. We first propose a RGBt Sampling strategy to sample red, green and blue channels at different times to extract local dynamics. Then we design a tokenization method to use different dimensional tubes to embed richer temporal information into the tokens. We also present the MinBlock architecture to implement a bio-inspired nonlinear connection between neurons to make the neurons more selective, we here extend MinBlock from image processing to video processing and we also explore the best position to insert MinBlocks within a UniFormerV2 architecture.

3.2.1 RGBt Sampling

Videos are high dimensional data and have different lengths, which means processing the entire videos at once is not affordable for computers and impossible for networks. Therefore, the strategy is to temporally downsample the videos into clips of the same length. For almost all Video Transformer networks, the inputs are obtained by sampling a certain number of frames from each video to form a clip representing this video. Usually, a video is uniformly divided into a certain number of segments along the temporal dimension, and then a frame is randomly selected from each segment. All the selected frames together form a clip to represent the original video. This sparse sampling strategy is reasonable because videos are proven to be redundant and have high temporal correlation (Tong et al. 2022).

However, due to the limited computational resources, the number of used frames representing a video is usually not enough and increasing the number of selected frames will improve the network performance. Especially for temporal-related datasets, the improvement will be more obvious. To obtain a good trade-off between network performance and computational costs, we introduce a novel strategy to sample the three color channels at different times. In detail, we first sample 3*N frames from each video rather than the original N frames. We then only use the R_{i-1}, G_i, and B_{i+1} channels

of the selected frames instead of using R_i, G_i, and B_i from the same frame, in order to form a RGBt frame containing temporal information, as shown in Fig 3.1. Thus, we not only introduce additional temporal information without increasing the input size but also reduce the spatial redundancy of frames to some degree.

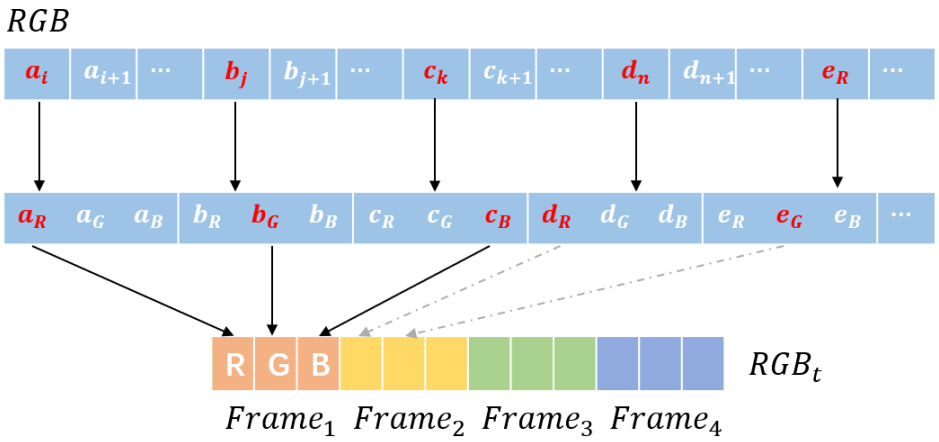
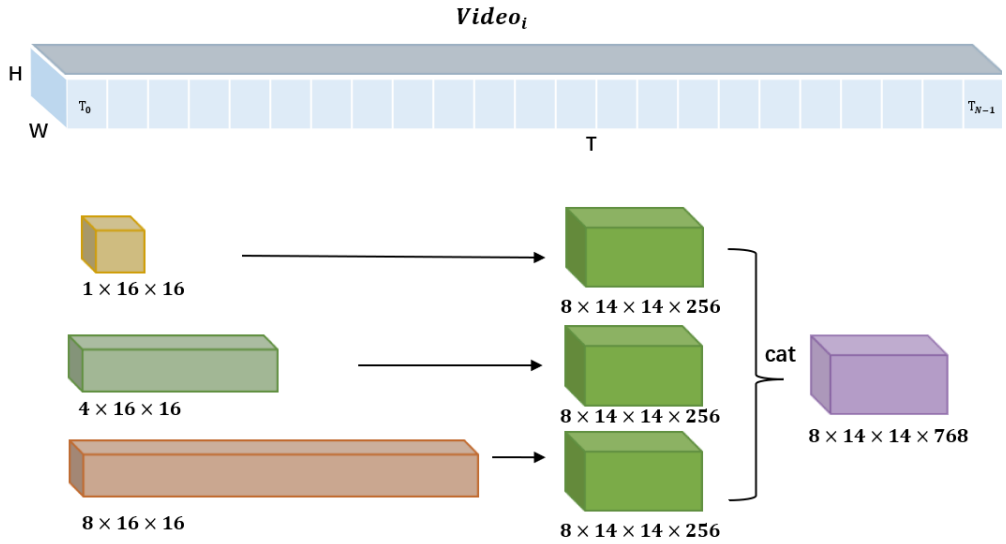


Fig. 3.1 Description of RGBt sampling strategy

3.2.2 Variable Sized Tubes Tokenization

Our backbone UniFormerV2 is one of the state-of-the-art architectures. It achieves impressive top-1 accuracies largely due to the pretrained Clip-ViT (Radford et al. 2021) weights. This benefits the network but also introduces some shortcomings. For using the pretrained weights of ViT, UniFormerV2 has to keep the same structure as ViT and then insert its own blocks into the ViT structure. Because of that, the 3D convolutional layer for tokenization in UniFormerV2 has to keep the same channel dimension of 768 as the ViT structure, which is very redundant. Inspired by the concept of Inception networks, we can fill 768 channels by concatenating different feature maps in the channel dimension by tokenizing video clips using 3D kernels of different sizes. By doing this, temporal information from frames of different lengths is fused by using tubes of different sizes. The obtained tokens can span different temporal periods (shorter or longer) and contain richer information about the dynamics of the actions.

Fig. 3.2 shows how we implement our idea, we first sample 32 frames from each video to form a clip with dimension 32 × 224 × 224 × 3 (T × H × W × C) as input. Then we use tubes of three different sizes (1 x 16 x 16, 4 x 16 x 16 and 8 x 16 x 16) to tokenize the inputs, and obtain three outputs with the same shape: 8 × 14 × 14 × 256 (T_1× H_1 × W_1 × C_1). After that, we obtain the final tokens by concatenating the three outputs in the channel dimension, which have the same channel dimension of 8 × 14 × 14 × 768 as the tokenization layer of ViT.



Tubes	Kernel	Stride	Padding	Output
1	1 × 16 × 16	4 × 16 × 16	0 × 0 × 0	8 × 14 × 14 × 256
2	4 × 16 × 16	4 × 16 × 16	0 × 0 × 0	8 × 14 × 14 × 256
3	8 × 16 × 16	4 × 16 × 16	2 × 0 × 0	8 × 14 × 14 × 256

Fig. 3.2 Tokenization with tubes of different sizes

3.2.3 Bio-inspired MinBlock

Min-Nets are a variant of FP-Nets (Grüning et al. 2022) inspired by end-stopped cortical cells with units that output the minimum of two learned filters (Grüning and Barth 2022, Grüning and Barth 2023). In previous studies, it has been proven that the addition of MinBlocks can improve the performance of state-of-the-art CNNs in object recognition tasks and make the CNNs more robust (Grüning et al. 2022, Grüning and Barth 2023). ENets are networks that employ the bio-inspired principle of end-stopping, and both FP-Net and Min-Nets are particular variants of eNets (Grüning and Barth 2023). Here, we combine not two units, but three pairs of two units in order to generalize eNets from images to videos. Such computations are related to optical flow computation (Barth 2000*b*) and also to the way biological neurons process motion information (Barth and Watson 2000, Barth 2000*a*). The geometrical motivation is based on the fact that the curvature of a 3-dimensional manifold defines the structure of the manifold and is captured by the invariants of the Riemann curvature tensor based on the sum of 3 pairwise combinations of the derivatives (Barth 2000*b*, Barth and Watson 2000).

Technically, our MinBlock consists of three point-depth-wise convolutional layers, the pairwise minimum operations and the add operation, as shown in Fig 3.3. We insert three additional 1x1x1 convolutional layers to convolve the previous feature maps and

use minimum functions to element-wise combine the feature maps learned by each pair of 1x1x1 convolutions, and we use an add operation to combine the three outputs from minimum operations. The 1x1x1 convolutional layers are used as spatiotemporal filters and capture features across channels by creating one-to-one projections of the feature maps. The pairwise minimum operations aim to make the neurons more selective and more robust than classical neurons (Grüning and Barth 2022). And the add operation is to keep the final output dimension the same as the input of MinBlock.

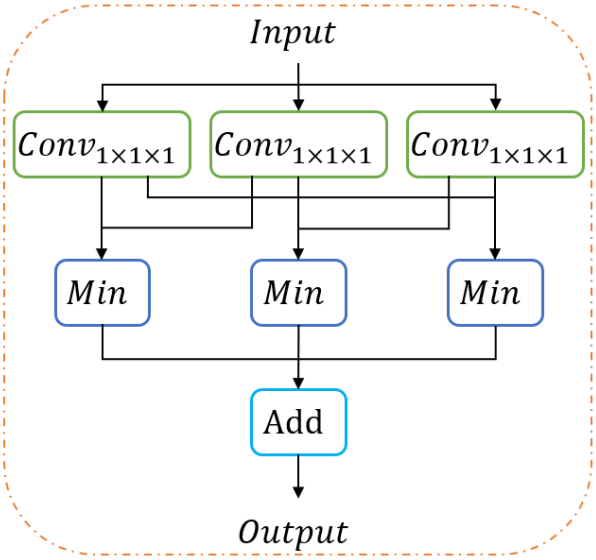


Fig. 3.3 The structure of a MinBlock.

There is clear evidence that MinBlocks can improve the performance of convolutional neural networks on image processing. Therefore, we expect MinBlocks to bring performance improvements in video understanding tasks as well. To investigate the inserted position of MinBlocks, we insert MinBlocks at two different locations in the backbone architecture that performs convolutions. As shown in Fig 3.4, one position we chose is after the 3D convolutional tokenization layer is performed. And the other position is inside the Local UniBlock as shown in Fig 3.5.

3.3 Experiments

3.3.1 Datasets

Since we lack the computational resources to deal with larger benchmarks, we use the classic scene-related dataset UCF101 and a subset of the temporal-related dataset Something-Something V2 as our datasets to verify the effectiveness of our novel design ideas.

UCF101 (Soomro et al. 2012*a*) is a small dataset for action recognition tasks. UCF101

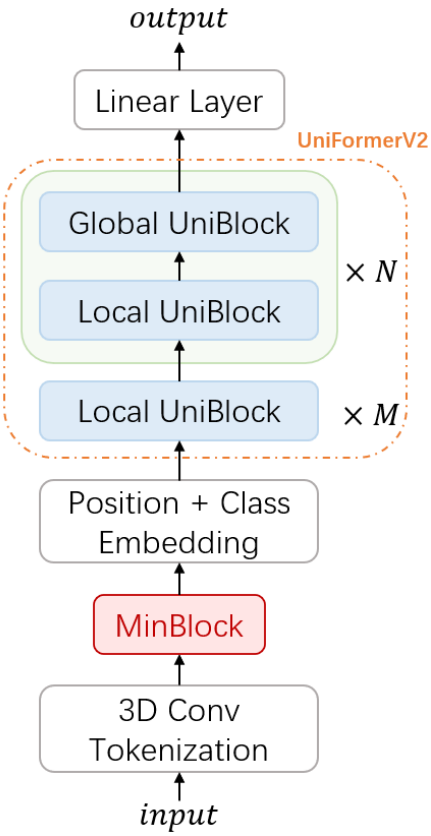


Fig. 3.4 Inserting MinBlock after the tokenization layer.

has 101 different action classes with mainly five types of actions: human-object interaction, body-motion only, human-human interaction, playing musical instruments, sports. Those actions are rather defined by spatial information, which means a relatively good performance can be obtained even without any temporal information. UCF101 has 13320 videos with an average length of 7.21s and a spatial resolution of 320x240 pixels, it consists of around 9.5k training videos and 3.7k validation videos. We randomly select videos proportionally from each class in the training set forming a total of about 1.6k videos for our validation set, and use the original validation videos as our test set.

Something-Something V2 (Goyal et al. 2017) is one of the most popular datasets for Video Transformers in video understanding tasks. It consists of 220,847 videos, with around 169k in the training set, 25k in the validation set and 27k in the test set. The actions in Something-Something V2 are more temporal-related and thus require more temporal information for making correct predictions. It has 174 different fine-grained action classes of human-object interaction scenarios, with an average duration of 4.03s. More specifically, Fine-grained means the understanding of the actions relies on videos rather than images, which is important for validating the capacity of networks in capturing temporal information. Action groups in Something-Something V2 usually contain some very similar actions with only subtle differences, and in order to distinguish these

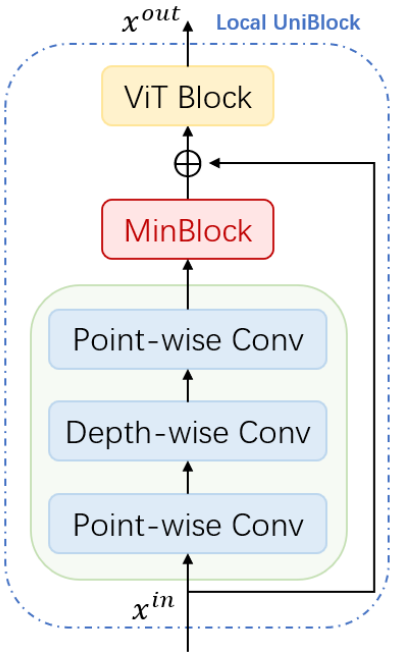


Fig. 3.5 Inserting MinBlock inside the local UniBlock.

similar actions within a group, a good understanding of the fine-grained actions at both spatial and temporal levels is required.

SthSth32 is a subset of the Something-Something V2 dataset containing only 32 classes selected to reduce computational costs. The training set, validation set, and test set are randomly grabbed from the original sets as suggested by Goyal et al. (2017). The resulting dataset SthSth32 contains about 41k training videos, 6.1k validation videos and 6.2k test videos - see Appendix B for more details.

We train our networks from scratch, because pretrained weights are trained on standard RGB frames and are not available for our RGBt frames. Unfortunately, both UCF101 and SthSth32 datasets are not large enough to train Transformer networks to maximum performance, we here therefore focus on the relative improvements of network performance brought by our novel design ideas.

3.3.2 Implementation Details

As described above, we first uniformly sample a number of frames to form a video clip representing the video. We sample 3*N (N is 8 for UCF101 and 4 for SthSth32) frames for RGBt sampling, 4*N frames for differently sized tubes tokenization and N frames for MinBlock. All selected frames are then resized into a jittering scale range [240, 320] and randomly cropped to 224x224 pixels for training. All frames are directly resized to 224x224 pixels for validation and test. Note that before feeding into the Transformer structure, the dimensions are the same: 8x224x224x768 (T × H × W × C).

Our training batch size is 256, both validation and test batch sizes are 128. We

choose the AdamW optimizer to learn network parameters and follow the training recipe reported in (Li et al. 2022*b*), a cosine learning rate schedule (Loshchilov and Hutter 2016) with a linear warm-up strategy for the first 5 epochs. Our warm-up start learning rate is 1e-6 and cosine end learning rate is the same, and the base learning rate we use is 1e-5. The momentum is set to 0.9 and the weight decay is 0.05. We conduct all experiments on 4 NVIDIA A100 40G GPUs.

3.3.3 Results and Discussions

We validate our three novel design ideas on UniFormerV2 backbone respectively using both the UCF101 and SthSth32 datasets. The improvements of network performance shown in the tables below demonstrate the effectiveness of our designs.

The results for the UCF101 dataset are shown in Table 3.1. The RGB row shows the baseline performance of the original UniFormerV2 with standard RGB frames, with 43.67% top-1 accuracy and 69.84% top-5 accuracy. RGBt sampling achieves a 3.2% higher top-1 accuracy compared to the RGB baseline without increasing the corresponding FLOPs and the number of parameters. And tokenization with tubes of different sizes with standard RGB frames obtains a 6.85% high top-1 gain in accuracy with slightly more FLOPs and parameters. And inserting MinBlocks into the backbone leads to a 2.34% top-1 accuracy improvement. We find that RGB with variable sized tubes setting improves the network the most based on both top-1 and top-5 accuracy.

Table 3.1 Comparison of RGB, RGBt, RGB tubes and MinBlock on UCF101.

Method	**#Frames**	**Param.(M)**	**FLOPs(G)**	**Top1**	**Top5**
RGB	8	123.82	157.41	43.67	69.84
$\mathrm{RGB_t}$	3*8→8	123.82	157.41	46.87	74.23
RGB tubes	4*8→8	125.78	160.50	50.52	79.54
RGB Min*	8	145.08	190.71	46.01	72.83

[1] 3*8→8 means 3*8 frames are used to form 8 RGBt frames;
[2] 4*8→8 means 4*8 frames are used for tokenization, and 8 is the temporal dimension after tokenization;
[3] Min* reports MinBlock inserted inside Local UniBlock.

Table 3.2 shows the results for the SthSth32 dataset. The baseline performance of the backbone with RGB frames is 44.91% top-1 accuracy. With RGBt sampling, the top-1 accuracy is improved by 5.75% compared to the RGB baseline without increasing parameters and FLOPs. And using different tubes with RGB frames leads to a 6.77% higher top-1 accuracy gain. The top-1 accuracy obtains 1.3% improvement by adding

MinBlocks. Again, we notice that RGB tubes setting achieves the best performance on both top-1 and top-5 accuracy.

Table 3.2 Comparison of RGB, RGBt, RGB tubes and MinBlock on SthSth32.

Method	#Frames	Param.(M)	FLOPs(G)	Top1	Top5
RGB	4	123.76	78.72	44.91	77.49
$\mathrm{RGB_t}$	3*4→4	123.76	78.72	50.66	81.94
RGB tubes	4*4→4	125.73	80.26	51.68	83.14
RGB Min*	4	145.03	95.37	46.21	79.02

[1] 3*4→4 means 3*4 frames are used to form 4 RGBt frames;
[2] 4*4→4 means 4*4 frames are used for tokenization;
[3] Min* refers to MinBlocks inserted inside the Local UniBlock.

The results from the Table 3.1 and Table 3.2 show that all our three designs can improve the performance for not only scene-related dataset but also temporal-related dataset (see overview in Fig. 3.6). The performance improvements indicate that using RGBt sampling can help the network to model longer dynamic dependencies of videos without increasing FLOPs and the number of parameters, adding Tubes with different sizes can embed richer temporal information into tokens and inserting MinBlocks presumably makes the neurons more selective of useful information than the classic neurons.

We also perform ablation experiments on UCF101 dataset to explore the best position for inserting MinBlocks. As discussed in previous section, MinBlocks can improve the performance of convolutions. Thus, we choose two positions with convolutions: one is after the 3D convolutional tockenization layer (as shown in Fig. 3.4), the other is inside the Local UniBlock and on top of ViT Block (as shown in Fig. 3.5).

Table 3.3 Overview of comparison of MinBlocks with different positions on UCF101.

Method	Position	Param.(M)	FLOPs(G)	Top1	Top5
RGB Min	Token	125.59	160.19	43.93	70.08
RGB Min	Local	145.08	190.71	46.01	72.83

[1] Min indicates the use of MinBlocks;
[2] Token: MinBlock inserted after the tokenization layer as in Fig. 3.4;
[3] Local: MinBlock inserted inside the local UniBlock as in Fig. 3.5.

Table 3.3 shows the results of ablation experiments, the MinBlocks inserted after the tokenization layer with RGB frames lead to a 0.26% performance improvement, while

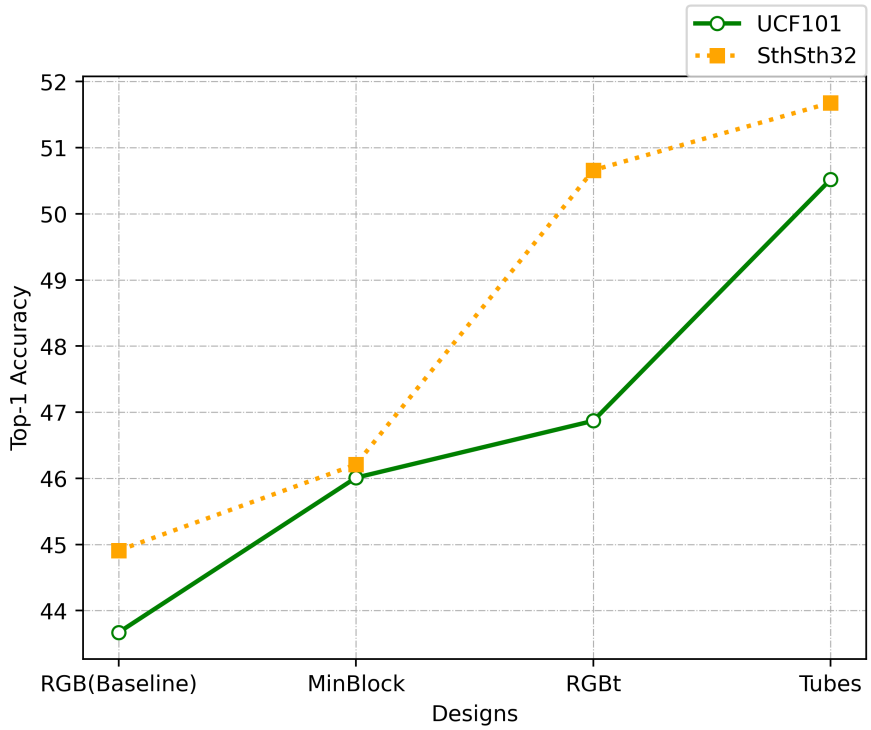


Fig. 3.6 Top-1 accuracy improvements by our designs on UCF101 and SthSth32.

MinBlocks inserted inside the local UniBlock obtain a 2.34% higher top-1 accuracy. In Table 3.1 and Table 3.2, we therefore report the better performances obtained by MinBlocks inserted inside the Local UniBlock.

Table 3.4 Extra experiments on UCF101.

Methods	Position	Top1	Top5
RGB and $\mathrm{RGB_t}$	-	48.30	74.25
$\mathrm{RGB_t}$ tubes	Token	53.22	79.60
RGB tubes and Min	Local	54.04	80.63
$\mathrm{RGB_t}$ Min	Local	49.08	75.42

With the inspiration of the Temporal Segments Networks (TSN) (Wang et al. 2016*b*), we used both RGB and RGBt frames as two different modalities and fused them at a late stage on UCF101 dataset to obtain a better performance. The first row of Table 3.4 shows that this fusion achieves a 4.63% top-1 accuracy improvement compared to using only RGB frames and a 1.43% top-1 accuracy gain compared to using only RGBt frames. From previous experiments we show that both RGBt sampling and tokenization with tubes of different sizes can improve the network performance, so that we also

try the combination of these two designs. The second row of Table 3.4 shows that the combination obtains a 9.55% top-1 accuracy gain relative to the RGB baseline and an extra 2.7% top-a accuracy improvement compares to adding only variable size tubes. Besides, we also run experiment on combining variable tubes with MinBlocks. This operation leads to a 10.67% higher top-1 accuracy than RGB baseline. Moreover, the combination of RGBt sampling and inserting MinBlocks also achieves higher accuracy (49.08%) than both only RGBt sampling (46.87%) and only MinBlocks (46.01%).

The performance of all combinations as well as the performance of individual designs are shown in the Fig. 3.7. Obviously, all kinds of pairwise combinations of our design elements can further improve the performance of the network.

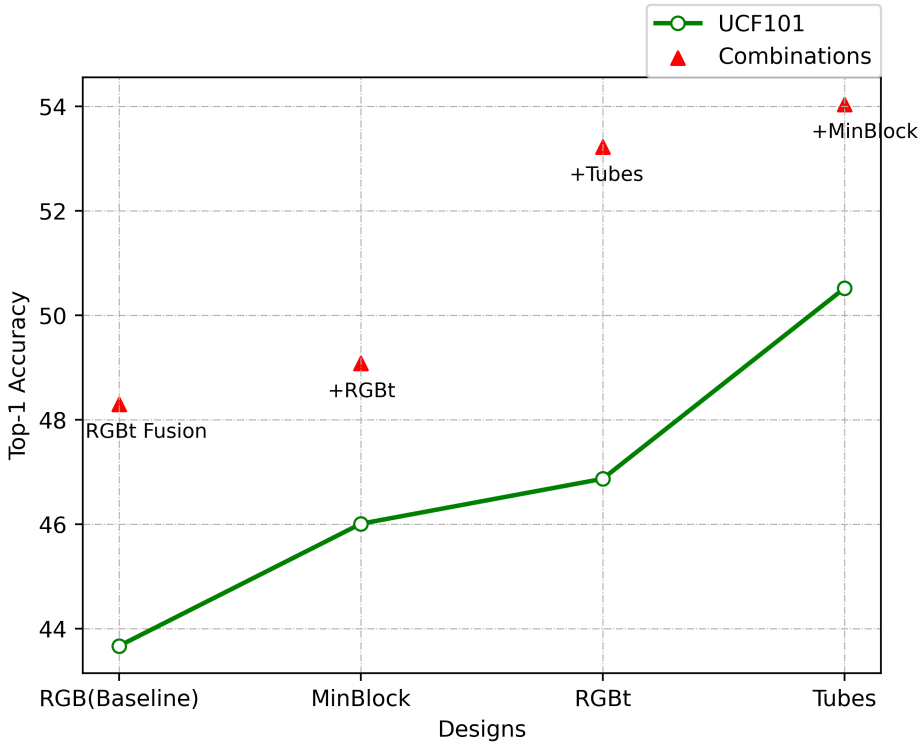


Fig. 3.7 Top-1 accuracy improvements by different combinations on UCF101.

3.4 Conclusion

In this chapter, we propose three novel design ideas for Vision Transformers in video understanding tasks. The first idea is to sample frames differently, which we called RGBt sampling, that is, sample 3 times the number of frames and select red channel, green channel and blue channel from the consecutive frames to form new RGBt frames. This way, the input clips not only contain three times longer temporal sequence but also maintain the same dimension, meaning that RGBt frames do not introduce additional computational costs and parameters. The second idea is to tokenize the input clips

with tubes of different sizes (with 3D kernel dimensions of 1x16x16, 4x16x16 and 8x16x16), in order to span the temporal information from different lengths of frames and thus embed richer temporal representation into the final tokens. The third idea is using MinBlocks to introduce a novel type of neurons which are more information selective and can make the network more robust. We also conduct ablation experiments to explore the best position to insert the MinBlocks.

We validate our three designs on a scene-related dataset and a temporal-related dataset. All results on both datasets show that our three novel ideas can improve network performances in action recognition tasks. Moreover, the combinations of two arbitrary design elements can further achieve even better network performance.

Due to the limitation of computational resources, we train our networks from scratch and focus only on the relative improvements compared to the baseline. We would expect better results on large datasets and using pretrained weights.

CHAPTER 4

Salient Spatiotemporal Slices on 2D-CNNs for Video Understanding

4.1 Introduction

Video understanding is a popular research field in computer vision. Action recognition and hand gesture recognition are two main classification tasks in video understanding. The purpose of action recognition is to capture spatiotemporal features that can represent the entire video and thus make predictions of the specific action. There are a lot of applications of action recognition, such as intelligent traffic control, smart home control and so on. For gesture recognition, the goal is to recognize hand gestures of different subjects, and virtual reality, augmented reality and human-computer interaction are common applications.

2D-CNNs have proven to be good at capturing spatial features of images but cannot capture temporal information. Therefore, for video understanding, researchers extended 2D kernels to 3D to also extract temporal information by adding a time dimension. However, 3D-CNNs can only model local dependencies in a rather small 3D neighborhood due to the limited receptive field. This limited 3D-CNNs performance in video understanding, since long-range dependency modeling is important for this task.

Video understanding remains a challenge because how to effectively use temporal information is still a key issue. Current methods generally sample a number of frames to form a video clip as an input representing the entire video. Such temporal downsampling often causes the loss of critical temporal information. Moreover, current state-of-the-art networks (e.g., visual transformer variants) require high computational costs even with the downsampled videos. Generally, video is treated as stacked images in time. In our work, we have a new perspective on video, treating it as a 3D block. If we look at the 3D block from the front, we see xy slices (frames); from the above perspective, we see xt slices and from the left we see yt slices. So, xt slices contain the horizontal spatial and temporal information and yt slices contain the vertical spatial and temporal information. The visualization of temporal information on slices makes it possible for 2D-CNNs to directly extract spatiotemporal features.

It is known that videos are redundant (Tong et al. 2022), because the spatial content changes little in consecutive frames. Thus, not all frames are needed for videos analysis.

Likewise, not all xt and yt slices are needed. The redundancy of spatiotemporal slices is more serious than that of spatial frames. Thus, we propose a simple method to exclude redundant xt and yt slices by using saliency.

Modeling longer range temporal dependencies can improve the network performance in video understanding, but requires higher computational loads. As most current methods are not able to utilize the temporal information across the entire video and our salient spatiotemporal xt and yt slices contain the complete timeline of the video, we propose a simple model that can capture the long-range spatiotemporal representations by using simple 2D-CNNs on our spatiotemporal slices with a low computational cost. Moreover, we evaluate our model on five different datasets (for both action recognition and hand gesture recognition tasks) to prove its efficiency and robustness. We conduct experiments on only xy, xt and yt slices respectively to compare the performances on different types of slices, so that we can evaluate the effectiveness of spatiotemporal slices. Moreover, we combine different types of slices to further improve performance.

4.2 Methodology

4.2.1 Spatiotemporal Slices (xt and yt)

Video is essentially frames stacked in time, so we can think of a video as a 3D cube in x, y, and t coordinate systems. In this perspective, a frame is a slice of the video in the xy plane (xy slice), which represents 2D-spatial information of the video. Thus, a slice in the xt plane (xt slice) represents horizontal spatiotemporal information of the video. Similarly, the slice in the yt plane (yt slice) contains vertical spatiotemporal representation of the video. All kind of slices are shown in Fig. 4.1. Xt and yt slices describe the movement trajectories of the subjects over the complete timeline of the video, i.e., containing every time index. In this way, action dynamics are well represented in the xt and yt slices, so that simple 2D-CNNs have capacity to extract spatiotemporal information simultaneously. 2D-CNNs trained on these spatiotemporal slices are of lower complexity compared to 3D-CNNs and Video Transformers.

4.2.2 Sampling Strategies

Videos have proven to be quite redundant, because the visual content changes slowly over consecutive frames. The redundancy also exists in xt and yt slices. We can take advantages of these redundancies by selecting only a subset of frames and slices that are useful and sufficient for action recognition. We aim to find sampling strategies that preserve the essential visual content and temporal context over the entire video.

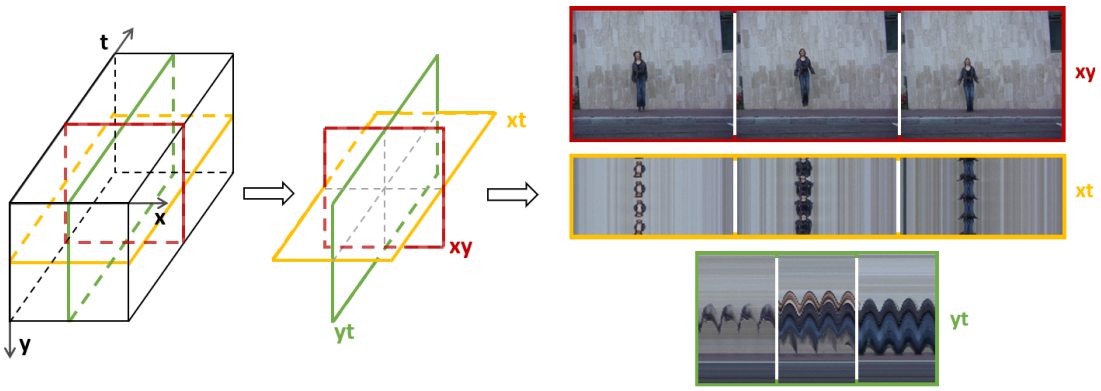


Fig. 4.1 Overview of slicing xy, xt, and yt slices ("Jump on place" action from the Weizmann dataset. Note how the jumping action is well represented in the spatiotemporal yt slices).

4.2.2.1 Sparse Sampling

For xy slices (frames) selection, we use a sparse global sampling strategy described in (Wang et al. 2016*a*). We first uniformly divide a video into N segments, and then randomly choose a frame from each segment to form a clip (defined by the selected frames) that represents the entire video. Uniform sampling helps our network to model spatial information changes of the entire video and random sampling makes our network more robust. However, this down-sampling of xy slices may lead to the loss of temporal information, that's why we need xt and yt slices to provide the information of the complete timeline.

4.2.2.2 Saliency-based Sampling

In most cases, for the subjects who are taking the action, their moving does not occupy the whole video cube; actions only take place at a certain region (as shown on xy slice in Fig. 4.2). So, there are some xt and yt slices without any motion trajectories. Objects that do not move, are represented as straight lines in the xt and yt slices (see non-salient slices shown in Fig. 4.2). These non-salient slices are not helpful for decision making or may even hurt the performance of the network. Thus, we calculate a saliency value for each slice to exclude such redundant slices.

Since the redundant frames are defined by straight lines, we use a simple curvature measure to detect the salient slices as those that do not contain only straight lines.

Assume we have a gray-scale image (or slice) I. If we take a patch at (μ, ν) and shift it by (x, t), the gray-scale differences of these two patches is

$$E(x,t) = \sum_{\mu} \sum_{\nu} \omega(\mu,\nu)[I(\mu+x,\nu+t) - I(\mu,\nu)]^2, \tag{4.1}$$

with $\omega(\mu, \nu)$ being a window that slides over the image.

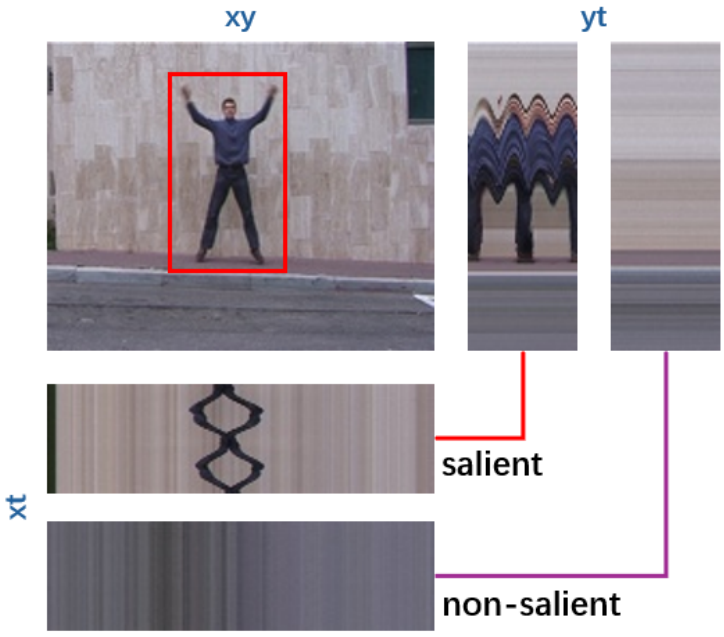


Fig. 4.2 Examples of salient slices and non-salient slices from the Weizmann dataset (Action: Jack). The salient xt and yt slices are defined by the region of interest indicated by the red rectangle.

After applying a Taylor expansion we can get

$$E(x,t) \approx \begin{bmatrix} x & t \end{bmatrix} J \begin{bmatrix} x \\ t \end{bmatrix}, \tag{4.2}$$

where J is the structure tensor (Jähne 1993)

$$J = \sum_{\mu} \sum_{\nu} \omega(\mu,\nu) \begin{bmatrix} I_x(\mu,\nu)^2 & I_x(\mu,\nu)I_t(\mu,\nu) \\ I_x(\mu,\nu)I_t(\mu,\nu) & I_t(\mu,\nu)^2 \end{bmatrix}. \tag{4.3}$$

When $\omega(\mu,\nu)$ is an identity matrix, J is simplified to

$$J = \sum_{x,t} \begin{bmatrix} I_x^2 & I_x I_t \\ I_x I_t & I_t^2 \end{bmatrix}. \tag{4.4}$$

A simple way to measure deviation from flatness is to use the determinant of J, which is equal to zero for straight lines:

$$R = det(J). \tag{4.5}$$

Technically, we first convert the xt and yt slices into gray-scale images and apply a Gaussian low-pass filter. Then we use the Sobel operator to calculate the derivatives in the x (and y) and t directions. After that, we calculate the different terms in Equa-

tion 4.4 and perform Gaussian filtering on these terms. Furthermore, we calculate the determinant of J to obtain R. Finally, we apply non-maximum suppression to get optimal values and use the average of these values for selecting salient slices. For fusing xy, xt, and yt features, we sample the same number of xt slices and yt slices by selecting the top values of calculated saliency. By using saliency-based sampling, we ensure that each xt or yt slice we select contains the entire motion trajectory of the subject.

4.2.3 Architecture

We use simple 2D-CNN architectures (such as the ResNet18) as backbones for our model. We first obtain all xy, xt and yt slices from videos and select the most informative slices by applying our sampling strategies as described in section 4.2.2, then we feed these selected slices into our CNN backbone. As shown in Fig. 4.3, we explore several ways to utilize the different types of slices for classification: (i) using the results obtained from only one type of slice (e.g. just xt slices), (ii) fusing two types of slices (e.g. xt and yt slices), and (iii) fusing all three types of slices (xy, xt and yt slices).

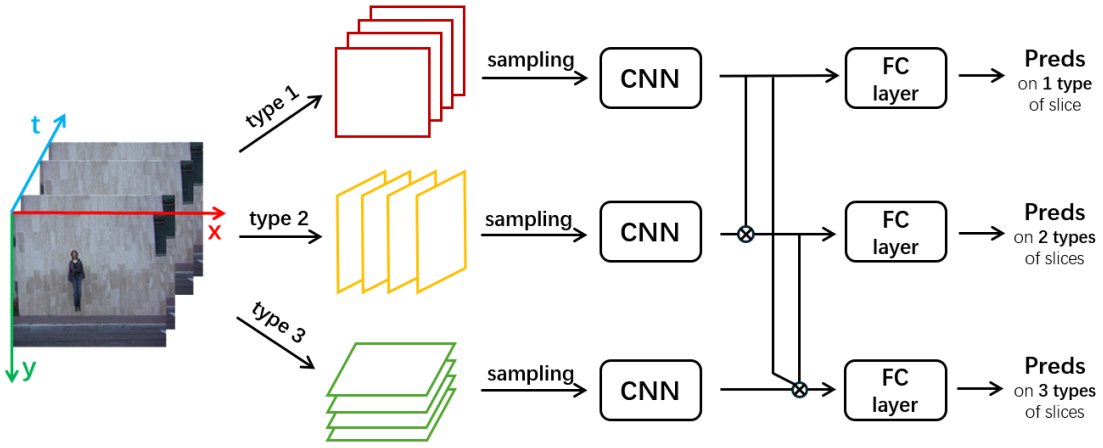


Fig. 4.3 Overview of our salient spatiotemporal slicing CNN. (Type 1, 2, 3 respectively represent one of xy, xt and yt slices; $\otimes$ means the combination of different types of slices.)

In Fig.4.3, the top branch describes the use of only xy, xt, and yt slices respectively. The results illustrate how useful the spatiotemporal slices are compare to xy slices. The middle branch implements different paired combinations of the three types of slices by using a two-stream network, because the spatiotemporal slices contain the temporal information of the entire video but cannot fully replace the visual content represented in xy slices. The last branch combines information from all three types of slices to explore complementary and redundancy of information.

Since our backbone is a 2D-CNN, the network makes decisions based on each individual slice. Thus, we use a voting mechanism to choose the majority prediction of the sampled slices from the same video as the final prediction for the video.

4.3 Experiments

4.3.1 Datasets

To evaluate the model, we use five rather small video datasets. Two for action recognition tasks and three for hand gesture recognition tasks, so that we can show that our network performs well not only on coarse-grained action recognition but also on fine-grained action recognition (hand-gesture). Datasets such as UCF101 (Soomro et al. 2012*a*) contain static visual clues of objects and backgrounds, i.e., actions can be recognized without much temporal information. Therefore we choose the Weizmann (Blank et al. 2005) and KTH (Schuldt et al. 2004) datasets (examples are shown in Fig. 4.4) in which the static frames do not contain many clues to reveal the action classes. Fine-grained activities are more difficult to distinguish, the reasoning relies more on the subtle differences over time. Therefore, we also use hand-gesture datasets such as the Cambridge Hand Gesture (Kim et al. 2007), Northwestern University Hand Gesture (Shen et al. 2012), and IPN Hand(Benitez-Garcia et al. 2021) datasets (examples are shown in Fig. 4.5). The classes of each dataset are detailed in Table 4.1.

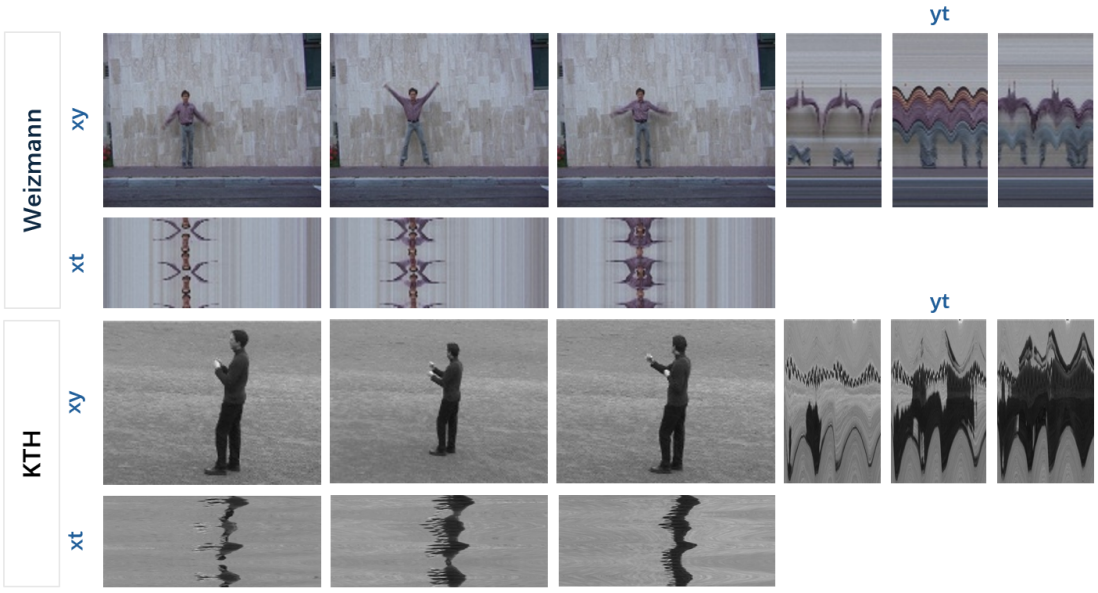


Fig. 4.4 Examples of frames and salient spatiotemporal slices of the action recognition datasets (Weizmann dataset with action *Jack* and KTH dataset with action *Boxing)*.

The Weizmann Dataset (Blank et al. 2005) consists of 90 videos (resolution 180x144, 25fps) collected from 9 different subjects, each subject performs 10 actions. And the backgrounds for all subjects and actions is the same.

The KTH Dataset (Schuldt et al. 2004) contains 2391 videos (resolution 160x120, 25fps) for 6 different actions, each action is performed by 25 different subjects within 4 different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors.

The Cambridge Hand Gesture dataset (Kim et al. 2007) includes 900 videos (resolution 320x240) of 9 hand gestures. These gestures are performed by two subjects and are generated by 3 different shapes and 3 motions under 5 different illumination conditions.

The Northwestern University Hand Gesture dataset (Shen et al. 2012) has 1050 videos (resolution 640x480, 30fps) of 10 hand gestures performed by 15 subjects. Each subject performs each hand gesture with 7 different hand postures, which largely increases the difficulty of classification.

The IPN Hand dataset (Benitez-Garcia et al. 2021) consists of 4039 videos (resolution 640x480, 30fps) with 13 hand gestures performed by 50 subjects. The gestures are generated in 28 different scenes and various subject-camera distances, which increases the generality of the dataset.

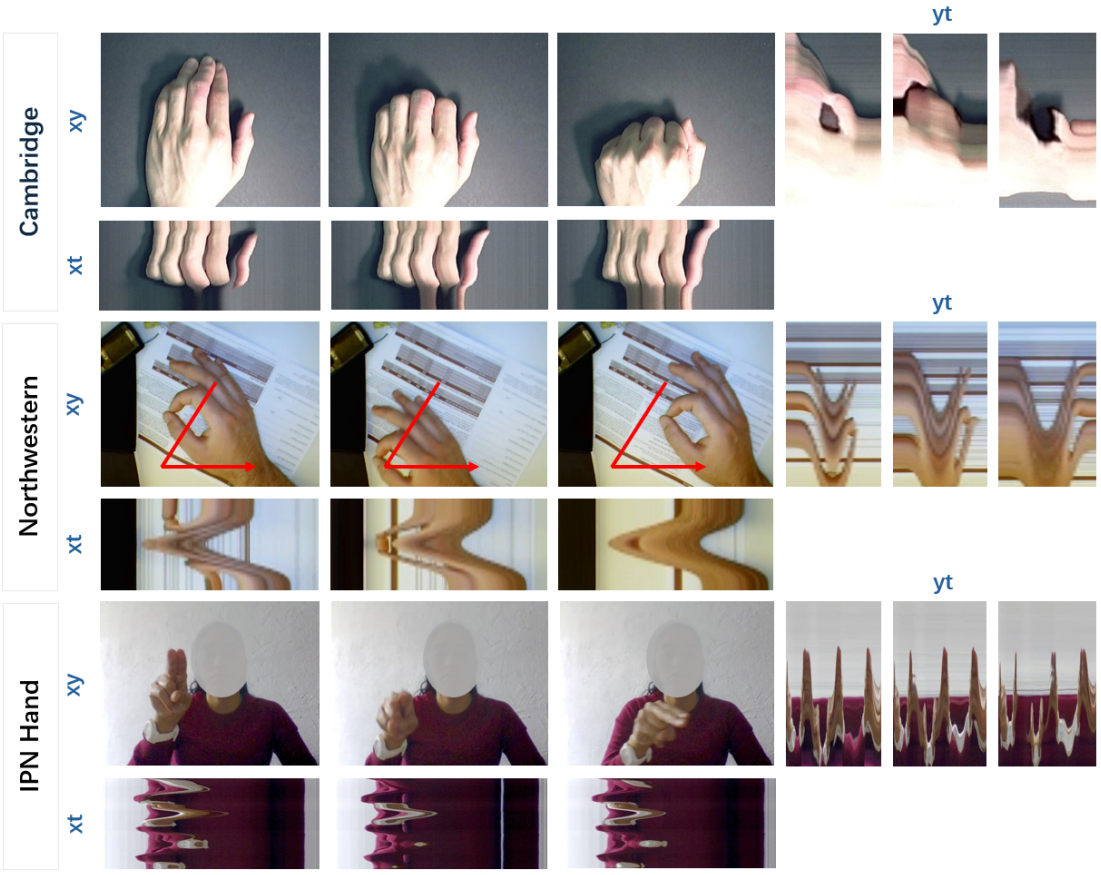


Fig. 4.5 Examples of frames and salient spatiotemporal slices from hand gesture-recognition datasets. Top: Cambridge Hand Gesture dataset with gesture *Contract from flat hand shape*. Middle: Northwestern University Hand Gesture dataset with gesture *Move down-right with "OK" hand shape (thumb and forefinger loop)*. Bottom: IPN Hand dataset with gesture *Pointing with two fingers*.

4.3.2 Implementation Details

We use the public train-test splits for the IPN Dataset and we randomly split the other four datasets into training set, validation set and test set with a ratio 6:2:2. We use a

Table 4.1 Classes of the five datasets

Dataset	#Class	Category
Weizmann	10	0:bending, 1:jack, 2:jumping, 3:jumping in place (pjump), 4:running, 5:galloping sideways (side), 6:skip, 7:walking, 8:waving two hands, 9:waving one hand
KTH	6	0:walking, 1:jogging, 2:running 3:boxing, 4:waving, 5:clapping
Cambridge	9	0:leftward(flat), 1:rightward(flat),2:contract(flat) 3:leftward(spread), 4:rightward(spread), 5:contract(spread) 6:leftward(v-shape), 7:rightward(v-shape), 8:contract(v-shape)
Northwestern	10	0:move right, 1:move left, 2:rotate up, 3:rotate down, 4:move down-right, 5:move right-down, 6:clockwise circle, 7:counterclockwise circle, 8:“cross”, 9:“Z”
IPN Hand	13	0:pointing with one finger, 1:pointing with two fingers, 2:click with one finger, 3:click with two fingers, 4:throw up, 5:throw down, 6:throw left, 7:throw right, 8:open twice, 9:double click with one finger, 10:double click with two fingers, 11:zoom in, 12:zoom out

ResNet18 pre-trained on ImageNet1K (Deng et al. 2009) as backbone for the Weizmann, KTH, and Cambridge datasets, and a ResNet34 pretrained on ImageNet1K as backbone for the Northwestern and IPN Hand datasets because the latter are larger and more complex.

We conduct all experiments on 3 NVIDIA GeForce RTX 2080 Ti GPUs. Both the training batch size and the test batch size are 128. The learning rate we use is 0.0001 and we train each experiment for only 20 epochs. For training, all slices are first resized to 256x256 and randomly cropped to 224x224 pixels. For testing, we directly resize all slices to 224x224 pixels.

4.3.3 Results and Discussions

Table 4.2 shows the results obtained for the action-recognition datasets. We find that the top1 accuracies obtained for xt and yt slices are better than those for xy slices on both datasets, which means that the motion trajectories in the salient spatiotemporal slices make the different actions more distinguishable. For the Weizmann dataset, yt slices are more useful than xt slices; for the KTH dataset, both xt and yt slices are helpful. These results indicate that our salient slices contain useful spatiotemporal information which enable 2D-CNN to capture temporal feature as well as spatial feature. However, the impact of horizontal and vertical spatial features may vary depending on the dataset. If

we combine different types of slices, the top1 accuracies can be further improved. The combination of xy and yt slices obtains 100% top1 video accuracy for the Weizmann dataset and the combination of xy, xt, and yt slices achieves 99.16% top1 video accuracy for the KTH dataset. The confusion matrices are show in subfigures (a) and (b) of Fig. 4.6.

Table 4.2 Top1 accuracy of different slices for action recognition on the Weizmann and KTH datasets.

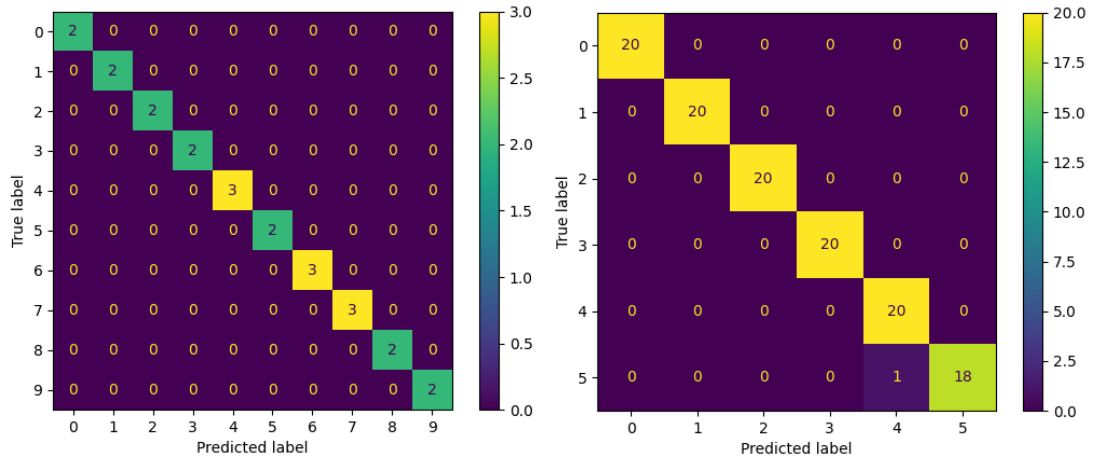
Datasets	**Weizmann**		**KTH**	
Slices	Frame Acc	Video Acc	Frame Acc	Video Acc
xy	67.66	73.91	77.94	83.19
xt	68.75	69.56	87.45	92.44
yt	80.43	86.96	87.39	92.44
xy + xt	80.43	78.26	91.12	92.44
xy + yt	90.46	**100.0**	91.12	96.64
xt + yt	82.61	<u>91.30</u>	93.33	<u>97.48</u>
xy + xt + yt	80.98	<u>91.30</u>	93.07	**99.16**

* Frame Acc means the accuracy for frames; Video Acc means the accuracy for videos after voting mechanism.
* **Bold** indicates the highest top1 accuracy, and <u>underline</u> indicates the second highest accuracy.

The results obtained for the hand-gesture recognition datasets are shown in Table 4.3. All performances of xt slices and yt slices are better than those on xy slices. As we described before, each gesture of the Northwestern dataset is performed with 7 different hand postures by different subjects, making it more difficult for the model to make correct predictions by using only the visual content of xy slices. This is the reason why the network obtains poor performances on xy slices (only 32.81% frame accuracy and 45.71% video accuracy). On the other hand, the top1 frame accuracies obtained on xt slices (88.62%) and yt slices (90.83%) are much higher than accuracy on xy slices, suggesting that the salient spatiotemporal slices contain relevant temporal features.

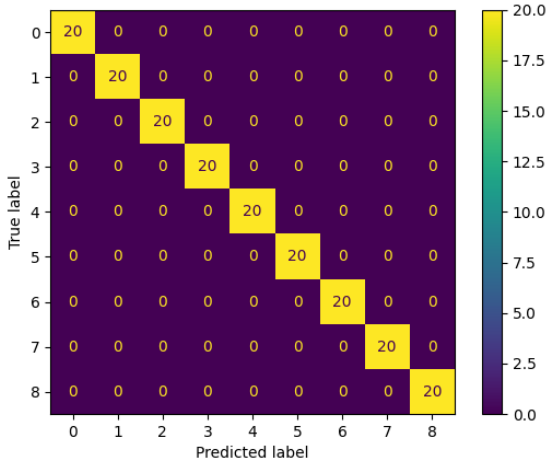
Similar to the results of the action recognition tasks, training the network on two or three different types of slices achieves better performance. The combination of xy and xt slices obtains 100% top1 video accuracy on the Cambridge dataset. As expected, using xy, xt, and yt slices leads to the highest accuracy (93.81%) on the Northwestern dataset; introducing xt and yt slices improves the top1 video accuracy by 48.1% compared to only xy slices. The model achieves 88.37% top1 video accuracy on the IPN Hand dataset by fusing the spatiotemporal features from xy and yt slices. Subfigures (c), (d), and (e) of Fig. 4.6 show the confusion matrices for these best models of different datasets respectively.

We find some interesting findings from the above results: (i) both xt slices and

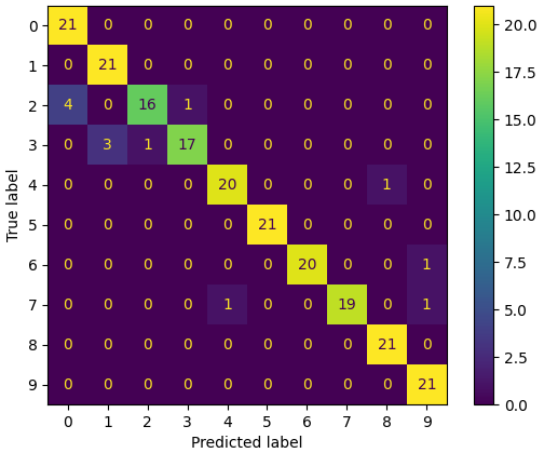


(a) Weizmann (xy+yt)

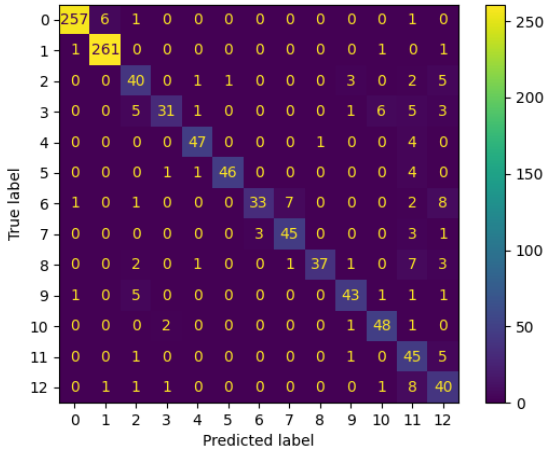
(b) KTH (xy+xt+yt)



(c) Cambridge (xy+xt)



(d) Northwestern (xy+xt+yt)



(e) IPN Hand (xy+yt)

Fig. 4.6 Confusion matrices of best model's performance for each dataset (The classes corresponding to the index values are shown in the Table 4.1).

Table 4.3 Results of different slices for hand gesture recognition on the Cambridge Hand Gesture, the Northwestern University Hand Gesture, and the IPN Hand datasets.

Datasets	**Cambridge**		**Northwestern**		**IPN Hand**	
Slices	F_Acc	V_Acc	F_Acc	V_Acc	F_Acc	V_Acc
xy	87.91	95.56	32.81	45.71	60.12	68.66
xt	97.48	97.22	88.62	89.05	67.90	71.30
yt	95.17	97.22	90.83	<u>91.90</u>	70.53	78.84
xy + xt	100.0	**100.0**	90.35	91.43	77.65	82.56
xy + yt	94.46	96.11	87.94	89.05	79.63	**88.37**
xt + yt	97.37	<u>98.89</u>	92.48	92.38	77.20	82.83
xy + xt + yt	98.53	<u>98.89</u>	91.43	**93.81**	81.81	<u>87.83</u>

* F_Acc means the accuracy for frames; V_Acc means the accuracy for videos after voting mechanism.
* **Bold** indicates the highest top1 accuracy, and <u>underline</u> indicates the second highest accuracy.

yt slices lead to better performances than xy slices (in our experiments, xt and yt slices always outperform xy slices, as shown in Fig. 4.7), (ii) the relative improvement brought by xt or yt slices can vary a lot (e.g. yt is more useful for the Weizmann dataset, but for other datasets the performances are similar on xt and yt), as it depends on the actions or hand gestures the dataset contains, (iii) the combination of different types of slices does not always improve the performance of the network (for instances, fusing yt slices with xy slices reduces the accuracy relative to only using yt slices on the Cambridge and Northwestern datasets and using all three types of slices does not always achieve the best performance).

The number of parameters and FLOPs for the two backbones that we use are shown in Table 4.4. As the numbers in the table show, our method is not computationally expensive and the network has few parameters. Moreover, our architecture is flexible since one can easily exchange the backbone network to adapt to specific datasets.

Table 4.4 Number of parameters (M) and Flops (G) for the two CNN backbones that we used.

Backbone	**ResNet18**			**ResNet34**		
#Slice Types	xy	xy+xt	xy+xt+yt	xy	xy+xt	xy+xt+yt
Parameters (M)	11.18	22.36	33.54	21.19	42.58	63.87
FLOPs (G)	1.82	3.65	5.47	3.68	7.36	11.03

We evaluate our model on five different datasets (both action recognition and hand gesture recognition tasks) and obtain very good results. Next, we would like to compare our results with those of other studies. The performances for five state-of-the-art

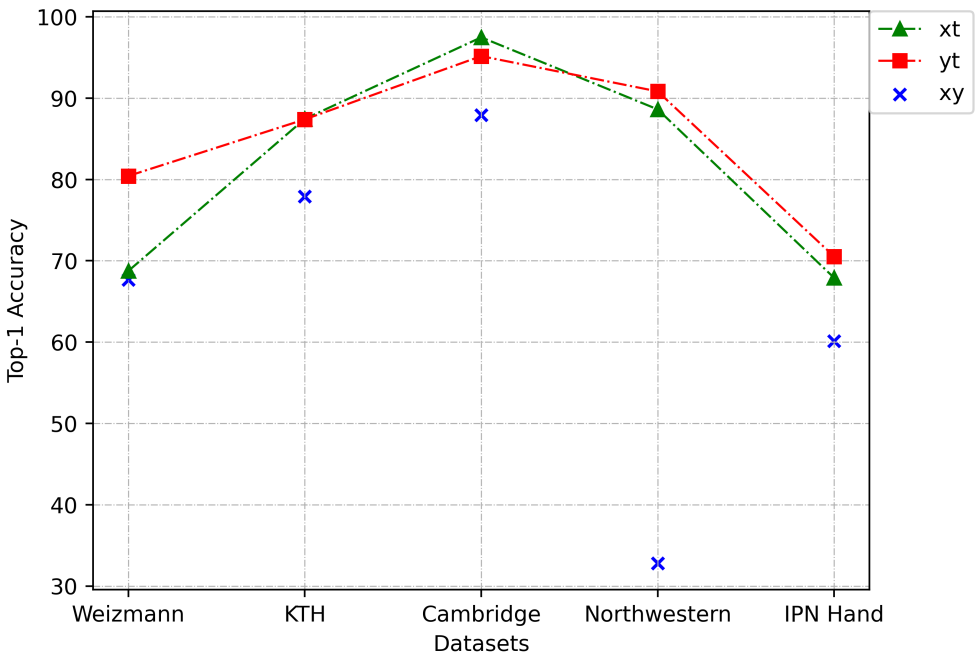


Fig. 4.7 Overview of the performances on xy, xt, and yt slices on different datasets.

methods on the Weizamnn and KTH datasets are shown in Table 4.5. Note that our approach outperforms all other methods, including those using optical flow and 3D-CNNs. Besides, the calculation of optical flow and the training of 3D-CNNs are much more expensive compared to our approach. Thus, our model achieves a good balance of performance and computational cost.

Table 4.5 Comparison with SOTA methods on the Weizmann and KTH datasets.

Methods	Weizmann	KTH
3D-ConVNet (encoded frames) (Baccouche et al. 2011)	94.58	94.39
2D-CNN (dynamic images) (Bilen et al. 2016)	85.2	86.8
3D-CNN (Gaussian-weighted aggregation) (Basha et al. 2022)	96.53	95.86
PCANet-2 (3 types slices) (Abdelbaky and Aly 2020)	100	90.47
AlexNet (optical flow) (Darafsh et al. 2021)	100	97.95
ours	**100**	**99.16**

Table 4.6 shows state-of-the-art results for hand gesture recognition datasets. The performances of our model surpass others on the Cambridge and IPN Hand datasets, but still need improvement for the Northwestern dataset. More importantly, our model obtains better results on the Cambridge dataset than 2D-CNNs combined with RGB frames and optical flow and also outperforms 3D-CNNs on the IPN Hand dataset, which once again prove that our model achieves a good trade-off between performance and computational load. The reason for why we do not obtain SOTA results on the North-

western dataset may be that the lengths of the videos are much shorter than in case of other datasets (mean length: 40 frames). Due to its high resolution (640x480), the information on the resized slices (680x40→256x256 or 480x40→256x256) that we feed into the network may have some degree of distortion and thus reduce performance.

Table 4.6 Comparison with SOTA methods on the Cambridge, the Northwestern and the IPN Hand datasets.

Methods	Cam	North	IPN
Key frames and feature fusion (Tang et al. 2019)	98.23	96.89	-
C3D (Tran et al. 2015)	-	89.36	77.75
ResNetXt-101 (RGB+OF) (Benitez-Garcia et al. 2021)	-	-	86.32
ResNetXt-101 (RGB+Seg) (Benitez-Garcia et al. 2021)	-	-	84.77
2D-CNN with key frames (RGB+OF) (Yu et al. 2022)	98.62	97.64	-
Two stream CNN with BGRU (RGB+OF) (Verma 2022)	99.4	**98.6**	-
ours	**100**	93.81	**88.37**

* OF means optical flow frames; Seg means hand segmentation frames.

4.4 Conclusion

In this chapter, we propose novel slices called salient spatiotemporal slices, which can represent the spatiotemporal information of videos and enable simple 2D-CNN backbones to effectively capture the relevant temporal information needed for video understanding. We process videos from a new perspective - not as stacked RGB frames, but as 3D blocks that can be sliced along different axes (x, y, and t axes) - this is a simple but effective approach for video understanding. The xt and yt slices can help visualize the dynamics of the videos, and can help the network to have a more intuitive understanding of different actions. Moreover, we introduce an easy way to exclude non-salient slices by measuring the deviation from straight lines with the determinant of the structure tensor.

We conduct a series of experiments on both coarse-grained action recognition and fine-grained hand gesture recognition datasets to illustrate the contributions of the different types of slices, and also the effect of using the combinations of different types of slices. We find that the results on spatiotemporal slices always outperform those on only xy slices and the combination of slices usually leads to an improvement in network performance. Moreover, our results suggested that a simple architecture with proper design can outperform state-of-the-art methods. We expect those good results of our network because 2D-CNNs are good at extracting local orientation of images and here we encode the dynamics of an action as local orientations in the xt and yt slices and feed them into our network.

In the future, we hope to design different backbones to find the one that best captures the temporal information of the entire video for better performance. Furthermore, we would like to find a solution to the distortion problem of the motion trajectories when resizing the slices of short high-resolution videos, in order to overcome current challenges.

CHAPTER 5

Medical Application: 2D-CNNs using Salient Spatiotemporal Slices to Analyze Glaucoma and Visual Impairment via Walking Patterns

5.1 Introduction

Glaucoma is a common eye disease, mostly found in the elderly. It causes visual impairments and even irreversible blindness, so early detection and diagnosis are crucial for intervention and treatment of glaucoma. Although there is no cure for glaucoma, timely intervention and treatment can prevent or slow down its progression. However, clinical diagnosis of glaucoma is complex and requires patients to take a lot of examinations and visual tests, which require clinical expertise and are time-consuming and expensive. In addition, insufficient medical resources are common in remote and underdeveloped areas. The development of artificial intelligence in the medical field provides the possibility of remote diagnosis of glaucoma and visual impairment, which can alleviate the current medical dilemma to a certain extent.

The increase in computing resources and large amounts of medical data has enabled the application of artificial intelligence methods, especially deep learning, in medicine and neuroscience and achieved good results. Specifically, some proposed convolutional neural network (CNN) models utilize medical images, such as retinal fundus photography and optical coherence tomography (OCT) for glaucoma diagnosis and prediction.

Artificial intelligence models usually use the same data as the clinical experts do, such as various visual field test results, clinical data, retinal images, and OCT images. A few studies examine the impact of glaucoma and visual impairment on motion and show there is evidence that glaucoma affects brain structure and function, leading to defects such as impaired visual-motor coordination. This may result in impaired postural control and orientation of the patient, thereby increasing the risk of falls. Fall is the leading cause of injury-related death among glaucoma patients. Walking becomes more difficult as people age and the risk of falls increases accordingly. Moreover, glaucoma develops with age and further affects peripheral or side vision, meaning that elderly patients with glaucoma are at further increased risk of falls due to some degree of vision loss and the effects of advanced age. Thus, we would like to study the relation of glaucoma, visual impairment, and motion.

CNNs have become the most successful deep learning networks in glaucoma analysis. Compared with fully connected neural networks, local connections and weight sharing strategies of CNNs enable the network to better optimize and suppress model overfitting. Compared with Vision Transformers, CNNs are more friendly to small medical datasets and limited computing power, and are easier to generalize. Therefore, we use CNNs as the backbone of our network.

There are two common motion capture (Mocap) systems, one is the marker-based optical motion capture systems, for example, using the Vicon system to capture the movement of markers worn by participants; the other is the inertial measurement unit (IMU)-based motion capture systems. However, both methods require professional equipment and specialized laboratories. In addition, subjects may feel uncomfortable wearing the markers or IMU. Therefore, we would like to design a neural network architecture that performs well on simple video recording. This way, we only need to place a camera at a fixed location to capture moving participants.

In our dataset (Beyer et al. 2024), elderly health controls and elderly glaucoma patients walked on a treadmill at different speeds while doing visual function measurements and clinicians recorded the experimental videos for further investigation. Although CNNs perform well in action recognition tasks, these actions often have significant differences that can be easily identified. The biggest challenge with CNNs in detecting glaucoma and visual impairment through motion is that even clinicians cannot find differences between them and controls by motion. The differences in motion between the two groups are subtle and not sufficient for classification using common CNNs.

Videos are images stacked in time (with x, y and z axes), and each frame has a frozen action at a time index. Since all participants perform the same action, it is difficult to distinguish glaucoma or visual impairment patients from healthy controls by learning the spatial features on frames. Therefore, we change the perspective, treat the video as a 3D cube and slice it. By slicing, we obtain xt slices and yt slices that contain spatiotemporal information. We refer to them as motion slices. In addition, the motion slice has the complete motion trajectory of the participant without missing any details. Hu and Barth (2024) proposed a novel 2D-CNN model to recognize not only coarse actions but also fine-grained gestures. In order to analyse the motion trajectories on the slices, we use their model to analyze glaucoma and visual impairment based on the motion patterns.

5.2 Methodology

Compared with motion data captured by Mocap systems and other clinical data, video recordings do not require specialized equipment and expertise, making them easier to

access and cheaper. However, video recordings are more difficult to analyze than typical clinical data or images. 3D-CNN and Video Transformer have achieved great success in video understanding tasks, but they both require large datasets and computing power. Unfortunately, our clinical dataset is small and we lack computational resources. Furthermore, the networks are not designed for motion analysis, and detailed motion information is easily lost during temporal downsampling of these networks. 2D CNNs are more friendly to small datasets and do not require high computational cost, and the motion slices enable them to use the entire motion trajectory in the video recording. So we use the proposed 2D-CNN architecture (Hu and Barth 2024) for motion analysis.

5.2.1 Pre-processing

The recorded videos have a resolution of 1920x1080 (horizontal) or 1080x1920 (vertical) pixels and a frame rate of 25 frames per second. Each video is approximately 60 minutes long. When recording the video, some participants took a break during the experiment, or there was some intervention from the experimental conductor. Therefore, we manually clip these distractors from the videos. We manually crop parts of the video that do not involve motion to reduce the computational load on the network. The dimensions of the new video are 1280x1080 pixels and 1080x1280 pixels. Since there are two kinds of videos, horizontal and vertical, we rotate the vertical videos to ensure that all the same kind of slices contain the same kind of spatiotemporal information (horizontal or vertical spatiotemporal). Since the videos are too long and participants' walking patterns are repetitive, we segment the videos into shorter clips. Each clip consists of around 264 frames ($\sim$10.56 seconds), which is close to the clip length of public video datasets. Afterwards, we resized the video from 1280x1080 pixels to 264x222 pixels to speed up training.

5.2.2 Motion Slices

Due to the redundancy of video and limited computing resources, a clip composed of multiple frames obtained through temporal downsampling is usually used to represent the entire video in most 3D-CNN and Vision Transformer networks. A lot of motion details are lost during temporal downsampling, which does not have much impact on coarse motion classification since excessive detail is not required. However, these details are crucial for detecting visual impairments using tiny anomalies in motion. 2D-CNNs lack the ability to model the temporal dependence between frames. Temporal features can only be extracted after being combined with a recurrent neural network (RNN) such as a long short-term memory network. But the length of time series that RNN can process is limited, and the network may miss some key clues for decision-making.

In order to obtain the entire motion trajectory contained in the video recordings, we use the spatiotemporal slices (xt and yt slices) described in Chapter 4 as motion slices. The examples of xy, xt and yt slices of our video recording dataset are shown in Figure 5.1.

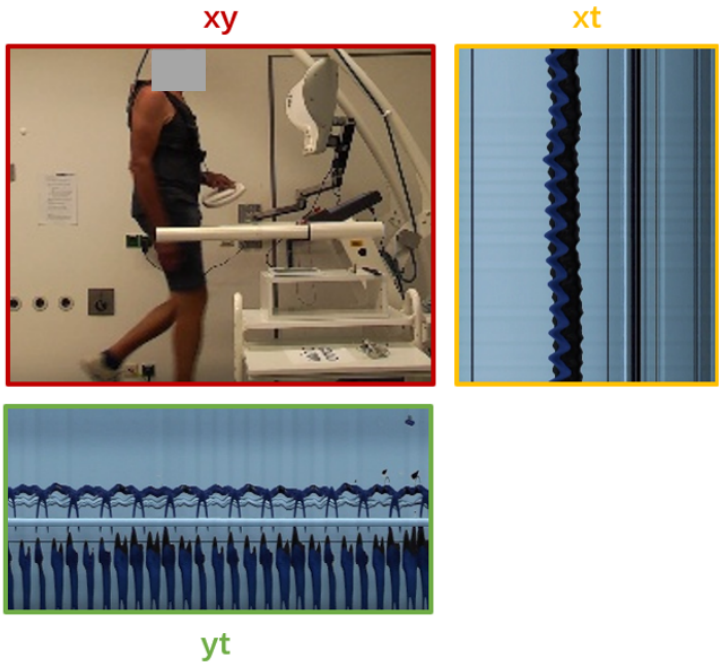


Fig. 5.1 Overview of slicing xy, xt, and yt slices (As we can see different motion trajectories on xt and yt slices).

5.2.3 Sampling Strategies

Sparse sampling for xy slices Since the motion is similar on each frame, there is some redundancy among xy slices. Therefore, we use sparse sampling to downsample the video clips. Specifically, we uniformly segment each video clip into N snippets and randomly select one frame from each snippet to form a new clip representing the original video clip.

Saliency sampling for motion slices There is also some redundancy among motion slices. We observe that the walking participants do not occupy the entire video block, which results in some slices not containing any motion trajectories. As Figure 5.2 shows, the non-salient slices obtained from parts of the video outside the red rectangle do not have any motion information, but only the static background. We need to exclude these non-salient frames as they have no useful information and may even harm the performance of the network. Only the salient slices obtained within the red rectangle will contribute to network decisions.

As described in Chapter 4, Section 4.2.2.2, we calculate saliency for each frame

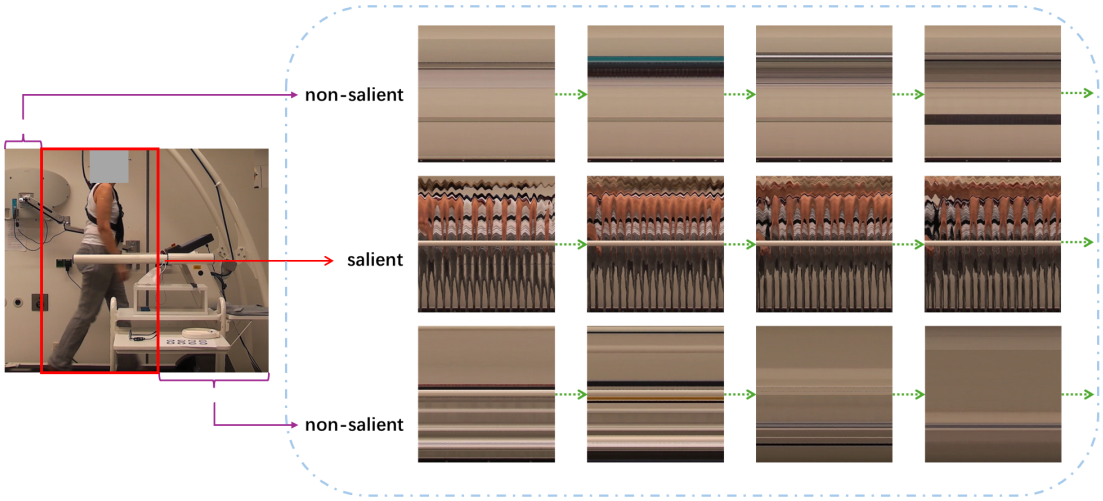


Fig. 5.2 Overview of yt slices from t_0 to t_{263}. By calculating the saliency of each slice, we can exclude non-salient slices and keep only salient slices.

and only the N slices with the highest saliency values are selected. For the method of calculating saliency, see Algorithm 1

Algorithm 1 Slice Saliency Calculation

Input: xt slice or yt slice I;

Output: The average saliency value of slice, S_{avg};

1: Convert slices into gray-scale image I_g;

2: Apply a Gaussian low-pass filter;

3: Use Sobel operator to calculate the derivatives in the m and n directions:
$dx = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} * I$ and $dy = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * I$;

4: Calculate terms I_{xx}, I_{yy} and I_{xy} for structure tensor J: $I_{xx} = dx^2$, $I_{yy} = dy^2$ and $I_{xy} = dx * dy$;

5: Perform Gaussian filtering on the above calculated terms;

6: Calculate the determinant $R = I_{xx} * I_{yy}$ - $I_{xy} * 2$;

7: Apply non-maximum suppression to get optimal values;

8: Calculate the average of the optimal values S_{avg};

5.2.4 Workflow and Architecture

We use the shallow network ResNet18 as our backbone for our simple architecture because our dataset is rather small. Our motion slices enable 2D-CNNs to extract spatiotemporal features directly from images, so complex architectures are unnecessary. It

is worth noting that the backbone of our architecture is replaceable and more suitable backbones can be used for specific tasks.

Figure 5.3 (a) and (b) show the processing of long videos and the detailed workflow of our algorithm. As described in Figure 5.3 (a), a raw video is clipped into several video clips, each video clip consists of about 264 frames. Then, a video clip is sliced differently to obtain xy, xt and yt slices. After that, sparse sampling is used to select N xy slices that are roughly evenly distributed in time and salient sampling is applied to select xt and yt slices with the N highest salient values to form a new clip representing each video clip. Next, the new clips are fed into our 2D-CNN model to make predictions for each slice. And we use the majority of the predictions of all slices in a new clip as the prediction for the corresponding video clip. Similarly, the predictions of most video clips in the same video represent the prediction of this participant. The detailed workflow is described in Figure 5.3 (b).

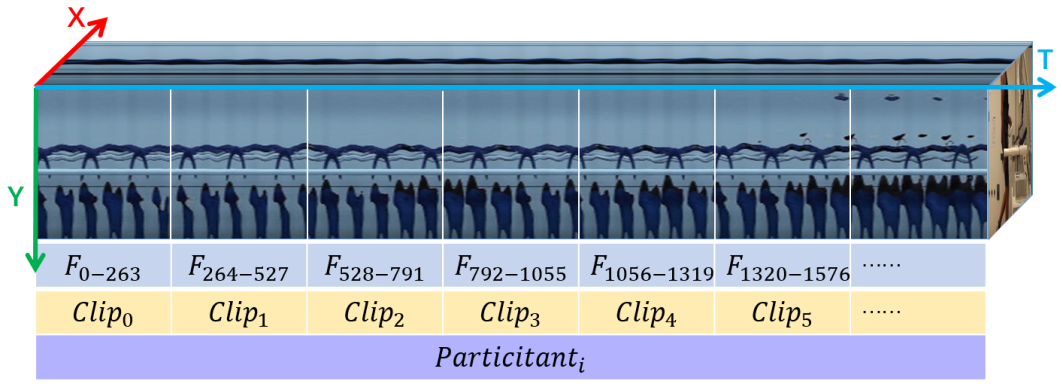
In order to obtain the best model for our task, we explore three different architectures as Figure 5.3 (c) shows. As shown in Figure 5.3 (c) left, we use a single-stream architecture to process different types of slices respectively and compare the performance of different slices. As described in Hu and Barth (2024), the combination of different slices can improve the network performance. So we also use a two-stream network, as shown in Figure 5.3 (c) middle and a three-stream network, as shown in Figure 5.3 (c) right to explore the best combination. In these two architectures, we fuse features of different slices at a late stage. We aim to obtain better results by combining different types of slices in our task.

5.2.5 Performance Metrics

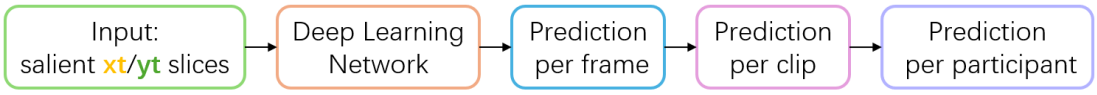
Performance metrics are used to evaluate the network performance and different fields of study often use specific evaluation matrices. In the medical field, accuracy, sensitivity and specificity are often used to evaluate a model, thereby we also use these three metrics for performance evaluation. Accuracy is the most popular evaluation matrix for classification. It indicates the proportion of correctly classified samples. However, accuracy does not describe the proportion of each class that is correctly classified, which is important in clinical research. Sensitivity is the proportion of correctly classified positive samples (healthy controls, labeled as 1) and specificity is the proportion of correctly classified negative samples (diagnosed glaucoma or visual impairment, labeled as 0). The accuracy, sensitivity and specificity are defined as below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5.1}$$

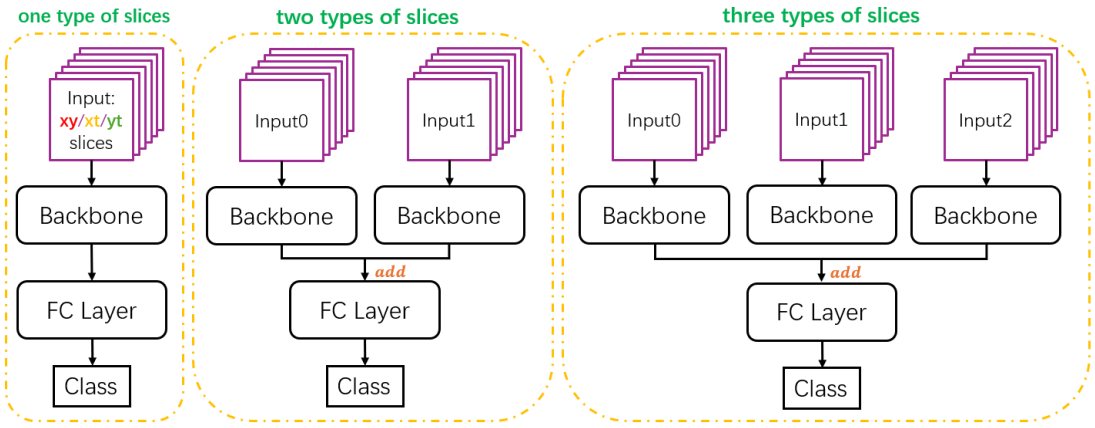
$$Sensitivity = \frac{TP}{TP + FN}, \tag{5.2}$$



(a) The processing of our long video. First, we divide a long video into several 264-frame snippets and obtain different types of slices; then we sample different types of slices from snippets to form clips based on our sampling strategies; all the clips obtained from one participant represent the participant.



(b) The workflow of the proposed approach. Predictions are for frame, clip and participant.



(c) The overview of our architecture. The left part shows the case for a single type of slice, the middle part shows the case for two types of slices, and the right part shows the case for all three types of slices.

Fig. 5.3 Detailed workflow and architecture of our network. **(a)** The processing of our long video. **(b)** The workflow of our proposed approach. **(c)** The overview of our architecture.

$$Specificity = \frac{TN}{TN + FP}, \tag{5.3}$$

where TN is the number of true negatives, FP is the number of false positives, TP is the the number of true positives and FN is the number of false negatives.

5.3 Experiments

5.3.1 Datasets

Our participants include 18 glaucoma patients and 30 healthy controls, all aged over 60 years. The dataset consists of video recordings of these participants walking on a treadmill while they perform three visual tasks (visual field, visual acuity and contrast sensitivity tasks) under three conditions (static, dynamic and dynamic+ conditions). Static refers to participants standing in place, dynamic refers to participants walking at a speed of 3.5 km/h, and dynamic+ refers to participants walking at their preferred speed.

5.3.2 Implementation Details

For video preprocessing, we use the ffmpeg package to crop and clip videos and use h264 video encoder to change the video format to mp4 format. Since our dataset is rather small, we use 5-fold cross validation to evaluate our model. We randomly spilt the participants into 5 folds and these folds are the same across tasks and under different conditions for better comparison. We choose the ResNet18 pre-trained on ImageNet1K (Deng et al. 2009) as backbone for our architectures.

We conduct experiments on one GeForce RTX 3060 GPU. Our training batch size and validation batch size are both 128, and we set the learning rate to 0.0001. For each experiment, we train the model for only 10 epochs. For training, all slices are first resized to 256x256 pixels and then randomly cropped to 224x224 pixels. For validation, all slices are directly resized to 224x224 pixels.

5.3.3 Results and Discussions

5.3.3.1 Diagnosed Glaucoma vs Healthy Control

Our dataset is highly unbalanced, with almost twice as many healthy controls as glaucoma patients. Besides, glaucoma patients vary greatly in severity, meaning they are in different stages of glaucoma, this makes the dataset even more imbalanced. To verify if there are significant differences in motion patterns between diagnosed glaucoma patients and healthy controls and to reduce the impact of dataset's imbalance, we randomly select 18 healthy controls to compare with 18 diagnosed glaucoma patients.

We first carry out experiments for the visual acuity task under three conditions and the results are shown in Table 5.1. We find that the accuracy of frame, clip and participant for the dynamic, dynamic+ and static conditions on xy, xt and yt slices are all around 50%, which indicate that there are no significant difference between diagnosed glaucoma patients and healthy controls in our dataset. And we also notice that all specificity values are quite low, meaning that it is difficult to distinguish diagnosed glaucoma patients from healthy controls. After we check the results of the different folds we use, we notice that performances vary a lot across folds. This finding again shows that the severity of glaucoma diagnosis varies greatly among patients. We recommend to group diagnosed glaucoma patients by severity in future data collections.

Table 5.1 Comparison of diagnosed glaucoma patients and healthy controls with different types of slices in Visual Acuity (VA) task.

	Dynamic			**Dynamic+**			**Static**		
xy	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	49.96	49.87	52.78	51.62	52.04	50.00	49.15	50.04	50.00
Sen %	59.23	59.84	66.67	59.40	62.10	61.11	65.53	66.86	66.67
Spe %	41.39	40.73	38.89	44.17	42.24	38.89	35.43	35.70	33.34
xt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	53.81	56.15	50.00	56.54	57.55	55.56	52.69	55.15	52.78
Sen %	65.84	70.10	66.67	66.71	68.35	66.67	66.45	71.47	72.22
Spe %	40.96	41.35	33.34	45.48	45.77	44.45	38.49	38.19	33.34
yt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	52.85	51.75	47.22	53.71	55.73	55.56	47.92	48.39	50.00
Sen %	67.35	70.08	61.11	64.71	66.76	61.11	59.30	62.51	66.67
Spe %	39.26	34.55	33.34	42.83	44.80	50.00	37.45	35.94	33.33

* Frame: results for frames; Clip: results for clips; Par: results for participants.
* Acc: accuracy; Sen: sensitivity; Spe: specificity.

5.3.3.2 Visual Impairment vs Healthy Control based on OCT

Although no differences in motion patterns are found between diagnosed glaucoma patients and healthy controls in our experiments, there are many studies assessing visual impairment through optical coherence tomography (OCT) by deep learning methods. Therefore, we next use our architecture to analyze differences in motion patterns between visually impaired subjects and healthy controls.

First, we have to assign visually impaired subjects and healthy controls. Since there are preperimetric glaucoma patients in our dataset, we perform a principal component analysis (PCA) using data including OCT to regroup the participants into visually im-

paired subjects and healthy controls, in order to examine whether there is a relationship between retinal changes and motion patterns through our model. Our PCA analysis is based on OCT measurements because they are sensitive to changes in retinal structure and retinal thinning is one of the early symptoms of glaucoma disease. We also include age and gender as covariates in our PCA analysis to eliminate the influence of irrelevant variables. The percentage of variance explained by the principal component 1 (PC 1) is 91.3%, thereby we use PC 1 to regroup. As shown in Figure 5.4, we choose 10 participants from each extreme side as visually impaired subjects (participants in the upper right box) and healthy controls (participants in the lower left box). Then we train our model again by using new groups for different visual tests under various conditions.

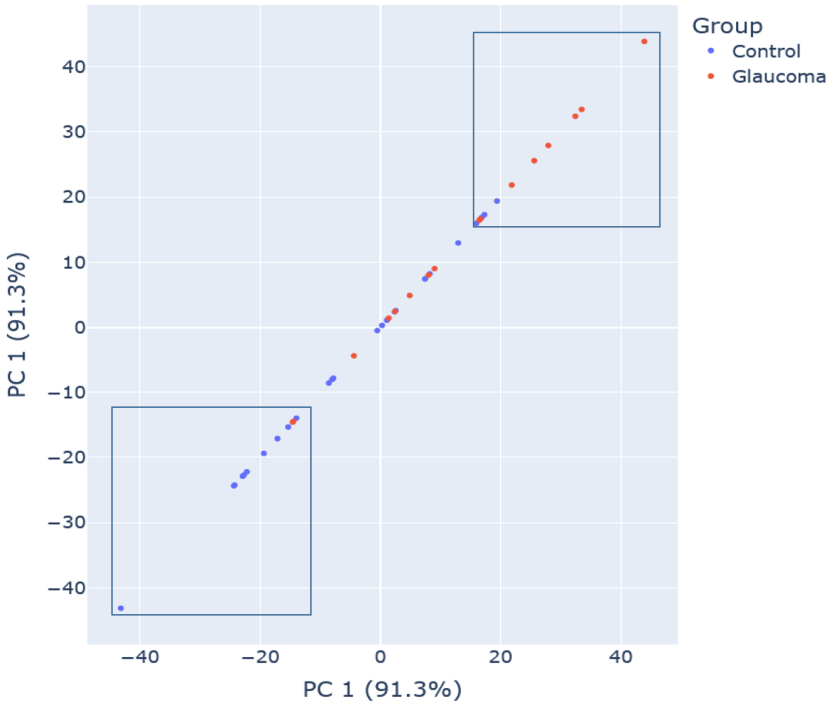


Fig. 5.4 The selection of participants using the first dimension of the PCA.

Similar to previous experiments, we first conduct experiments on the visual acuity task to test whether visually impaired subjects and healthy controls can be distinguished and Table 5.2 shows the related results. The accuracy under all conditions on all types of slices (except under the static condition on xy slices) suggests that there are strong differences in motion patterns between visually impaired subjects and healthy controls. We find that motion slices generally perform better than xy slices. Xt slices obtain the best results among three different types of slices, achieving the highest participant accuracy of 85% in the dynamic condition.

To further verify our above experimental observations, we then conduct experiments for the contrast sensitivity task and the visual field task. Table 5.3 shows the results for

Table 5.2 Comparison results of visually impaired subjects and healthy controls with different types of video slices in Visual Acuity (VA) task.

	Dynamic			**Dynamic+**			**Static**		
xy	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	60.47	62.02	70.00	60.85	60.41	70.00	49.89	49.38	50.00
Sen %	71.41	75.43	80.00	56.15	54.28	60.00	66.20	66.23	70.00
Spe %	53.36	52.73	60.00	65.03	65.65	80.00	36.74	35.63	30.00
xt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	71.62	75.69	85.00	73.26	77.92	80.00	67.01	72.96	80.00
Sen %	63.87	68.07	80.00	64.05	66.01	70.00	74.27	83.06	90.00
Spe %	80.41	85.30	90.00	80.64	88.01	90.00	61.52	65.20	70.00
yt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	66.61	69.38	75.00	62.66	65.36	70.00	60.59	64.05	75.00
Sen %	56.32	61.98	70.00	43.16	43.81	50.00	61.35	63.05	70.00
Spe %	78.59	79.51	80.00	79.91	84.85	90.00	60.94	66.02	80.00

* Frame: results for frames; Clip: results for clips; Par: results for participants.
* Acc: accuracy; Sen: sensitivity; Spe: specificity.

the contrast sensitivity task, and the results again show that visually impaired subjects and healthy controls can be distinguished by analyzing motion patterns. All performances obtained for motion slices are better than those for xy slices, and xt slices are most useful in classifying these two groups. The highest participant accuracy of 80% is obtained on xt slices under the dynamic condition.

The results for the visual field task are shown in Table 5.4. For this task, yt slices do not perform as good as for the other two tasks. However, xt slices still obtain the best performances, again demonstrating that visually impaired subjects are distinguishable from healthy controls.

By comparing all results from Table 5.2, Table 5.3, and Table 5.4, we notice that xt slices always perform better than yt slices, which suggests that the horizontal spatiotemporal information in the xt slice is important for discriminating visually impaired subjects and healthy controls compared to the vertical spatiotemporal information in the yt slice. This indicates that visual impairment may have a greater impact on people's horizontal movements than their vertical movements. In general, motion slices perform better than xy slices, and xy slices seem unable to capture the nuances of motion in static condition. This demonstrates the effectiveness of our motion slices. Note that the only difference in the slices is the motion, since all three types of slices are slicing from the same video. Specifically, the motion differences are the frozen action at a certain time index in the xy slices, the horizontal spatiotemporal motion trajectory in the

Table 5.3 Comparison results of visually impaired subjects and healthy controls with different types of video slices in Contrast Sensitivity (CS) task.

	Dynamic			**Dynamic+**			**Static**		
xy	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	60.82	61.84	65.00	49.48	49.56	55.00	47.66	47.22	55.00
Sen %	67.67	69.10	80.00	60.71	61.15	60.00	71.52	71.19	80.00
Spe %	52.45	52.93	50.00	41.25	41.86	50.00	28.30	27.74	30.00
xt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	73.46	77.36	80.00	63.33	65.04	70.00	63.02	64.84	70.00
Sen %	64.16	64.98	70.00	41.60	41.60	50.00	65.09	67.95	70.00
Spe %	82.53	88.80	90.00	82.65	85.38	90.00	62.60	64.07	70.00
yt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	63.37	70.07	70.00	59.52	59.56	65.00	52.22	52.85	65.00
Sen %	52.10	52.18	50.00	33.22	28.77	40.00	55.19	47.89	60.00
Spe %	73.58	85.80	90.00	83.11	86.92	90.00	53.67	62.85	70.00

* Frame: results for frames; Clip: results for clips; Par: results for participants.
* Acc: accuracy; Sen: sensitivity; Spe: specificity.

Table 5.4 Comparison results of visually impaired subjects and healthy controls with different types of video slices in Visual Field (VF) task.

	Dynamic			**Dynamic+**			**Static**		
xy	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	57.30	57.48	55.00	54.15	53.36	55.00	60.10	59.75	60.00
Sen %	33.63	32.59	30.00	45.09	43.68	50.00	72.63	72.61	70.00
Spe %	77.41	78.77	80.00	59.83	60.31	60.00	54.55	53.93	50.00
xt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	68.92	76.06	75.00	68.68	73.65	70.00	75.16	78.43	80.00
Sen %	54.25	59.56	60.00	62.32	66.76	60.00	72.05	78.57	80.00
Spe %	84.27	93.30	90.00	77.93	83.78	80.00	79.96	79.97	80.00
yt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	51.98	48.07	50.00	50.21	50.98	50.00	58.01	60.19	65.00
Sen %	39.62	32.93	40.00	34.86	33.96	40.00	58.74	66.13	70.00
Spe %	62.83	60.58	60.00	61.04	64.70	60.00	59.80	58.80	60.00

* Frame: results for frames; Clip: results for clips; Par: results for participants.
* Acc: accuracy; Sen: sensitivity; Spe: specificity.

xt slices, and the vertical spatiotemporal motion trajectory in the yt slices. These motion differences make the network performance differ across the three types of slices. According to our experimental results, xt slices are the best slices to classify different groups in our task.

By comparing the results of all tasks under the same conditions, we find that the differences in motion patterns between the two groups are more apparent when performing the visual acuity task. By comparing performance across all conditions in the same task, we observe that visually impaired subjects are generally easier to detect when participants perform visual tasks in the dynamic conditions. Interestingly, some of our results differ from what neuroscientists hypothesize, as we find significant differences in motion patterns of xt slices under static conditions as well. This is easily explained by the fact that our participants do not absolutely stand still during the task. Although they stand in place, in order to complete the corresponding visual test tasks, visually impaired subjects have to make some body movements to compensate for their visual impairment. Our experimental results show that motion slices can accurately capture these motions. We find that differences in sensitivity and specificity are generally smaller in the static condition than in the dynamic and dynamic+ conditions. Specificity is much higher than sensitivity in the dynamic and dynamic+ conditions.

5.3.4 Ablation Experiments

As suggested by Hu and Barth (2024), a combination of different types of slices can make it easier for the network to distinguish different groups. To verify whether the above approach is useful for our task, we further implement experiments on combinations of different types of slices for the visual acuity task, i.e. the combination of xt and yt slices and the combination of all three types of slices. The results from Table 5.5 show that the combinations of slices hurt the network performance, which means there is no positive information compensation among different types of slices. Information decoupling is more beneficial in detecting visual impairments in our task.

5.4 Conclusion

Glaucoma is one of the leading causes of visual impairment and brings a lot of inconvenience to patients' lives. Early detection and intervention of glaucoma can effectively prevent patients from irreversible vision loss. There are many clinical methods to diagnose glaucoma, but these methods usually require the full participation of clinicians, and are very cumbersome and time-consuming. Some studies proved that glaucoma affects movement. Therefore, studying the relationship between patients and healthy controls in motion also provides a potential method for diagnosing eye diseases. In this chapter, we propose a novel 2D-CNN framework to study whether there are significant

Table 5.5 Comparison results of visual impairment and healthy controls with different types of video slices in the Visual Acuity (VA) task.

	Dynamic			**Dynamic+**			**Static**		
xt+yt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	62.04	62.24	60.00	58.49	56.96	55.00	63.11	70.52	75.00
Sen %	23.97	21.05	20.00	19.04	12.37	10.00	63.02	67.02	70.00
Spe %	97.07	100.0	100.0	94.19	97.39	100.0	63.34	73.61	80.00
xy+xt+yt	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**	**Frame**	**Clip**	**Par**
Acc %	54.02	51.97	55.00	62.72	63.18	70.00	47.08	48.99	50.00
Sen %	16.99	11.70	20.00	24.47	23.59	40.00	36.26	35.09	30.00
Spe %	87.23	88.00	90.00	95.56	97.39	100.0	56.83	61.54	70.00

* Frame: results for frames; Clip: results for clips; Par: results for participants.
* Acc: accuracy; Sen: sensitivity; Spe: specificity.

differences in movement patterns between patients and healthy controls by using video recordings of participants performing visual tasks.

Based on our results for the visual acuity task under three conditions, we do not find any significant differences in motion patterns between glaucoma patients and healthy controls, so we conclude that diagnosing glaucoma through motion patterns is not possible. By analyzing experiments of visually impaired subjects and healthy controls in all tasks under all conditions, we find that visually impaired subjects can be easily distinguished from healthy controls by using xt slices. This suggests that there is a relationship between visual impairments and motion patterns, indicating that we can use motion differences to detect visual impairments.

Currently, we use the majority of predictions from clips of the same participant to represent predictions for that participant, and the majority of predictions from slices of the same clip to represent predictions for that clip. However, there may not be abnormal walking patterns within the 264-frame-clip period, or the abnormal clips may not be the majority of all clips for a participant. In the future work, we would like to propose a better method to obtain the prediction for the participant.

CHAPTER 6

Effective Use of Color and Temporal Information for Video Analysis

6.1 Introduction

Artificial Intelligence (AI) aims to solve problems, perform tasks and meet demands by perceiving, learning, understanding and reasoning like human or even outperform human beings. Neural networks such as Convolutional Neural Networks (CNNs) or Transformers are currently the most popular AI approaches which are inspired by human vision. CNNs, for example, perform well because they include an additional bias, which is inspired by vision and reduces the complexity of fully connected networks. We further investigate potentially useful biases inspired by human visual knowledge to improve the performance of video understanding.

Color is an important part of vision and has been proven to have a great effect on attention (MacKay and Ahmetzanov 2005, Pan 2010). The sensor cells in the retina include approximately 120 million rod cells and 6 to 7 million cone cells, which are distributed in different parts of the retina. Rod cells are more sensitive to light (Neves and Lagnado 1999). Cone cells mediate color vision. There are three types of cone cells, red, green and blue (long, medium and short wavelength) respectively, and they are most sensitive at around 565 nm, around 535 nm and around 420 nm respectively (Bull 2014).

The RGB color space used in cameras is designed based on the way the human visual system perceives color. Some professional cameras have three CCD or CMOS sensors, each for red, green and blue channel respectively. The more economical way is to combine a sensor with a single color filter to represent three color channels (Bull 2014). Most color image sensors use Bayer Filters with 50% green elements, 25% red elements and 25% blue elements. Green filters are more numerous because the human eye has a peak sensitivity around 555 nm in the green region of the spectrum (Snowden et al. 2012).

The length of the clips representing the videos affects network performance, because longer clips contain richer temporal information but require more computing power from the network. We here propose a novel color sampling strategy inspired by the human vision system. Specifically, we sample color channels from frames at different

times rather than using the traditional RGB frames. It turns out that trading color information for temporal information pays off. We aim to introduce richer temporal information by using our novel color-sampling scheme.

Here we first choose CNNs as our backbones. 2D-CNNs have been proven to be good at image processing but lack the capacity to extract temporal features, so we use a two-stream network adding a temporal stream to show that our sampling strategy can provide useful temporal information for 2D-CNNs. 3D-CNNs are able to capture spatiotemporal features in a small 3D neighborhood but unable to model long-range temporal dependencies for video analysis. Thus, we explore both single and two-stream architectures for 3D-CNNs and show that our sampling strategy benefits 3D-CNNs and the performance can be even further improved by fusing a temporal stream using our sampling strategy. We also use Transformers as our backbone, because they are better at capturing long-range temporal context but suffer a lot from high computational costs. Similar to 3D-CNNs, we also use single and two-stream Transformers to show that our sampling strategy introduces longer temporal information, thereby improving the network performance of Transformers.

Note that our approach does not introduce any additional computational costs for the three networks we use. Due to the novel sampling strategies we cannot use weights pre-trained on standard RGB frames, so we train all the networks from scratch. And we evaluate our methods on both UCF101 (Soomro et al. 2012*b*) and HMDB51 (Kuehne et al. 2011) datasets and demonstrate the effectiveness of our color sampling strategy by obtaining significant improvements for both datasets.

6.2 Methodology

In this section, we give a detailed description of our color-sampling strategies. We also give the overviews of our 2D-CNN, 3D-CNN and Transformer architectures as well as the two-stream architecture that fuses the outputs of two single streams. The two-stream network operates on standard RGB frames and novel frames that we propose using color sampling strategies respectively.

6.2.1 Color Sampling Strategies

The representation of a frame F at time t_i is as follows: P represents the pixel in one frame, W and H are the width and height of the frame:

$$F_{(x,y,t_i)} = \begin{bmatrix} P_{(0,0,t_i)} & P_{(0,1,t_i)} & \cdots & P_{(0,W,t_i)} \\ P_{(1,0,t_i)} & P_{(1,1,t_i)} & \cdots & P_{(1,W,t_i)} \\ \vdots & \vdots & \ddots & \vdots \\ P_{(H,0,t_i)} & P_{(H,1,t_i)} & \cdots & P_{(H,W,t_i)} \end{bmatrix}. \tag{6.1}$$

Each pixel P consists of three color-channel values:

$$P_{(x,y,t_i)} = \left[\begin{array}{ccc} P_{R(x,y,t_i)} & P_{G(x,y,t_i)} & P_{B(x,y,t_i)} \end{array} \right]. \tag{6.2}$$

C represents a video clip that consists of N frames obtained after temporal down-sampling, s is the temporal sampling stride.

$$C_{(x,y,t)} = \left[\begin{array}{ccccc} \cdots & F_{(x,y,t_{i-s})} & F_{(x,y,t_i)} & F_{(x,y,t_{i+s})} & \cdots \end{array} \right]. \tag{6.3}$$

Usually, a frame F at t_i in a clip C is defined as:

$$F_{(x,y,t_i)} = \left[\begin{array}{ccc} F_{R(x,y,t_i)} & F_{G(x,y,t_i)} & F_{B(x,y,t_i)} \end{array} \right]. \tag{6.4}$$

Since each frame has three color channels and there is a certain degree of redundancy in color information, the idea is to sample colors from different frames and fusing them into a single frame, so that a single frame contains temporal information of three frames. In order not to increase the computational loads, we keep the shape of the input the same. We sample $3 * N$ frames from a video rather than the original N frames. For each group of three frames, we retain the green channel of the current frame at t_i and replace the red channel of the current frame with the red channel of the previous frame at t_{i-s} and the blue channel of the current frame with the blue channel of the next frame at t_{i+s}. We refer to these new frames containing temporal information as RGB_t frames. Fig. 6.1 shows the detailed description of obtaining RGB_t frames. The RGB_t frame at t_i is defined as:

$$F_{(x,y,t_i)} = \left[\begin{array}{ccc} F_{R(x,y,t_{i-s})} & F_{G(x,y,t_i)} & F_{B(x,y,t_{i+s})} \end{array} \right]. \tag{6.5}$$

We also use only green channels, as the green channel proved to be more predominant and less noisy among the three channels of the video files. The process is similar to obtaining an RGB_t frame, except that we replace the red channel of the current frame at t_i with the green channel of the previous frame at t_{i-s}, and the blue channel of the current frame with the green channel of the next frame at t_{i+s}. The details for obtaining GGG_t frames are shown in Fig. 6.2. The GGG_t frame at t_i is:

$$F_{(x,y,t_i)} = \left[\begin{array}{ccc} F_{G(x,y,t_{i-s})} & F_{G(x,y,t_i)} & F_{G(x,y,t_{i+s})} \end{array} \right]. \tag{6.6}$$

We also obtain BBB_t and RRR_t frames in the same way. In the following sections, we use the term X_t as a shortcut for all these non-RGB frames.

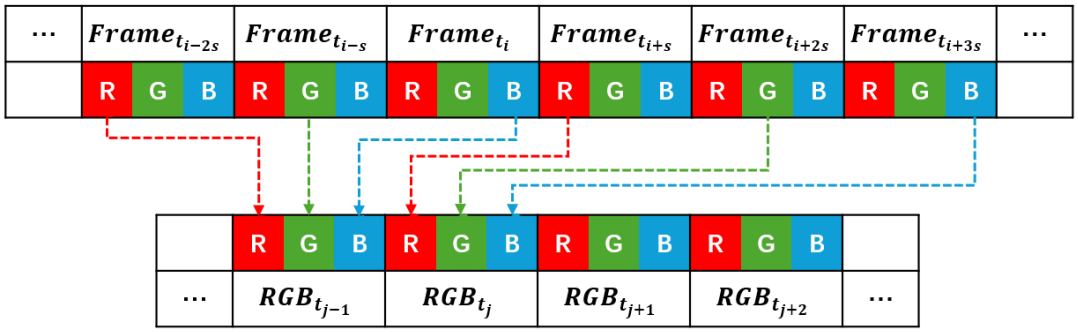


Fig. 6.1 Color sampling strategy for RGB_t frames.

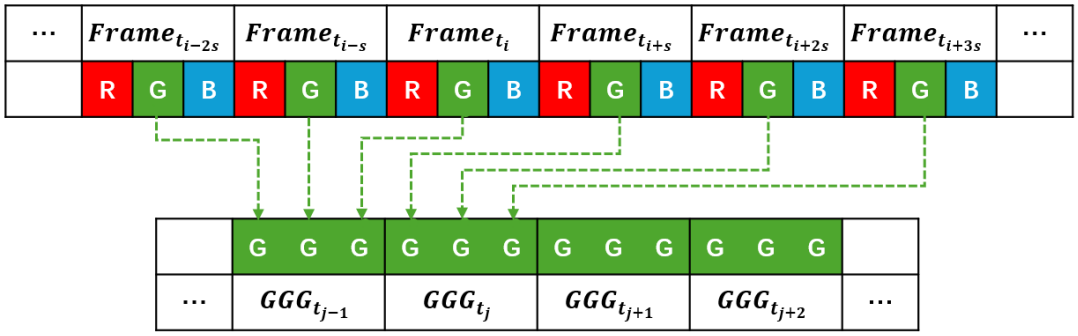


Fig. 6.2 Color sampling strategy for GGG_t frames. BBB_t and RRR_t frames are obtained in analogy.

6.2.2 Network Architecture

As we mentioned, current network performance is limited by the length of video clips due to computational constraints, and longer temporal dependencies help the network better model the context of the entire video and thus make more accurate predictions. To verify the capacity of our novel color sampling strategy to provide longer temporal information, we select 2D-CNN, 3D-CNN and Transformer as the backbones of our network.

We first propose a two-stream 2D-CNN network, where the spatial stream captures the detailed semantic features provided by the standard RGB frames and the temporal stream extracts dynamic patterns given by X_T frames that trade color information for temporal differences. By fusing these two streams, the network obtains spatiotemporal information that represents the entire video.

Fig. 6.3 shows our proposed 2D-CNN architecture. Because a 2D-CNN architecture can only make prediction for each frame, we average the predictions of all frames from the same video to obtain the final prediction for the video. We here choose a ResNet18 as our 2D-CNN backbone.

Then, we explore 3D-CNN architectures to further verify the effectiveness of our X_t frames. Unlike 2D kernels, 3D kernels can extract spatial and temporal features simultaneously. Thus, we use a single stream network to compare RGB frames with

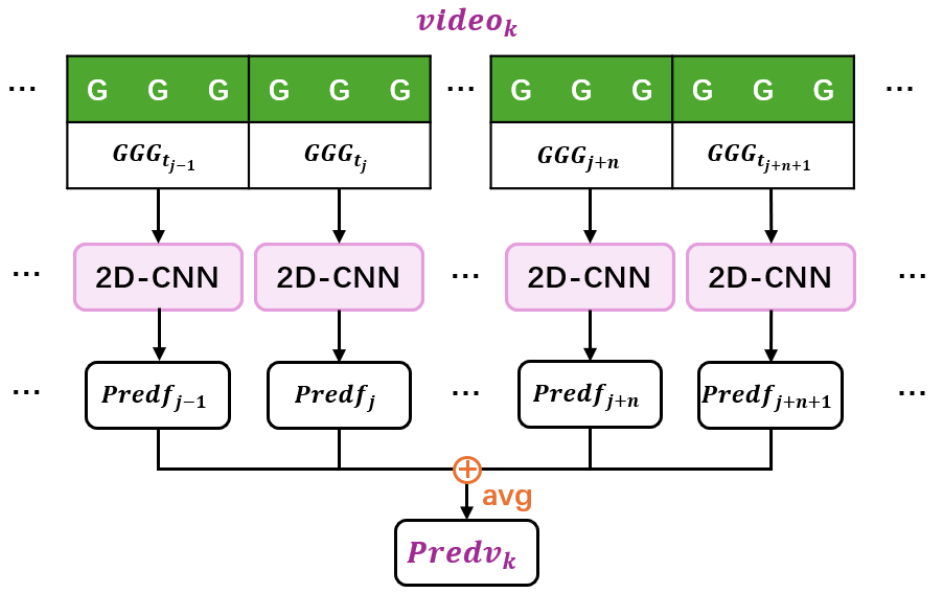


Fig. 6.3 2D-CNN architecture (here we use GGG_t as an example of input. $predf$ represents the prediction for each frame; and $predv$ means the final prediction of the video)

other X_t frames. Since we trade some color information for temporal information in X_t frames, we also use a two-stream 3D-CNN architecture for information compensation in order to obtain better performance.

Compared to 2D kernels, 3D kernels are inflated to an additional temporal dimension which enables 3D-CNNs to simultaneously extract spatiotemporal feature in a relatively small neighborhood. However, a 3D-CNN has limitations in capturing long-range temporal information due to limited receptive fields. A video clip consisting of X_t frames contains 3 times the temporal length of a clip consisting of RGB frames, so X_t frames can 'extend' the receptive field in the time domain to a certain degree. Thus, 3D-CNNs that operate on X_t frames can capture longer-range temporal context without increasing computational loads. Here we use a 3D-ResNet18 as backbone.

Transformers are designed to model global temporal dependencies of video clips, they are not limited by the receptive field like CNNs, but by the length of clips. However, Transformers require larger datasets and involve high computational costs. Increasing the length of a video clip to some degree can improve the performance of Transformers but it also leads to a higher computational load. Therefore, it is important to enrich the temporal information contained in a clip without increasing its actual length.

We also apply our X_t frames to single-stream and two-stream Transformer architectures. The input to the network composed of our proposed X_t frames contains richer temporal information and its size remains the same compared to using RGB frames. To demonstrate that our method also benefits larger and deeper networks, we choose UniFormerV2 as our Transformer backbone. The architectures for 3D-CNNs and Trans-

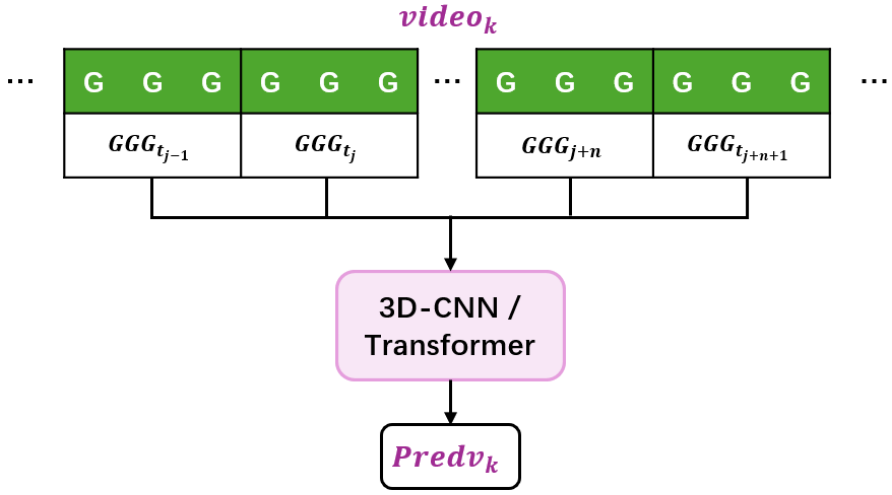


Fig. 6.4 3D architecture: 3D-CNN or Video Transformer as backbone (Here we use GGG_t as an example of input; and $predv$ means the final prediction of the video)

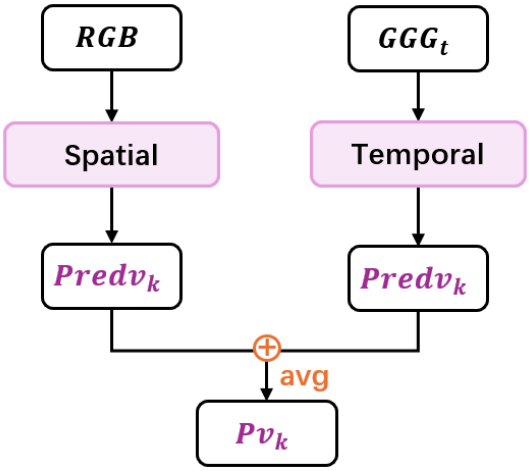


Fig. 6.5 Two-stream architecture: fusion of spatial and temporal streams.

formers are shown in Fig. 6.4.

We propose a two-stream network for the 2D-CNNs inspired by the SlowFast network (Feichtenhofer et al. 2019). For the 3D-CNNs and Transformers, we use a single-stream network and also a two-stream architecture inspired by I3D networks (Carreira and Zisserman 2017) to further improve the performance. By leveraging the two-stream architecture, the spatial stream focuses more on extracting spatial information from standard RGB frames, while the temporal stream operating on X_t frames provides additional temporal information for the network. We average the predictions of spatial and temporal streams as late fusion to obtain the final video predictions and this operation further improves the network performance. The two-stream architecture is shown in Fig. 6.5.

6.3 Experiments and Discussions

6.3.1 Datasets

Due to limitations in computational resources, we use two relatively small datasets to prove the effectiveness of our method: UCF101 and HMDB51. Both of them are classic benchmarks in action recognition tasks.

UCF101 (Soomro et al. 2012*b*) consists of 101 classes, with mainly five types of actions: sports, body-motion only, human-human interaction, human-object interaction and playing musical instruments. It contains 13320 videos with a spatial resolution of 320x240 pixels.

HMDB51 (Kuehne et al. 2011) has 51 classes of different actions that can be grouped in five types as well: general body movements, general facial actions, facial actions with object manipulation, body movements with object interaction, body movements for human interaction. It consists of 6766 videos with various spatial resolutions.

6.3.2 Implementation Details

We use a sparse sampling strategy for standard RGB frames. Firstly, a video is uniformly divided into N segments ($N = 8$ in our case), then a frame is randomly chosen from each segment to form a video clip representing the entire video. In case of X_t frames, we first sample $3 * N$ frames to represent a video using the same strategy as sampling RGB frames, and then fuse the temporal information from three consecutive frames in the clip into a single X_t frame as described in Section 3.2.

Like most studies, we use the public train-test split 1 for both datasets. Additionally, we randomly choose 12% of the videos from the training set as validation set for UCF101. All experiments are conducted on 4 NVIDIA A100 40GB GPUs. We set the training batch size to 256 for CNNs, 128 for Transformer, and the test batch size to 128 for all architectures. For network training, we first resize all frames to 256x256 pixels for the 2D-network or a scale jittering range [240, 320] for 3D-networks and then randomly crop them to 224x224 pixels. For inference, all slices are resized to 224x224 pixels for all architectures, and a single temporal clip with a random spatial crop (1x1 view) is used for 2D-CNNs and 10 temporal clips and 3 spatial crops (10x3 views) are used for 3D-networks.

6.3.3 Results and Discussions

Evaluation with 2D-CNNs. Since 2D kernels cannot capture spatial and temporal features simultaneously, we use a two-stream 2D-CNN architecture to evaluate the performance. It consists of a spatial stream operating on traditional RGB frames to capture spatial features and a temporal stream operating on proposed X_t frames to extract tem-

Table 6.1 Fusion results on ResNet18

Dataset	Modality	Frame Accuracy		Video Accuracy	
		Top1	Top5	Top1	Top5
UCF101	RGB	36.83	58.81	39.12	60.14
	$RGB + RGB_t$	38.37	53.00	40.52	62.33
	$RGB + GGG_t$	**39.06**	59.86	**41.55**	61.75
	$RGB + RRR_t$	38.41	61.28	40.59	62.94
	$RGB + BBB_t$	38.24	58.18	40.84	59.95
HMDB51	RGB	11.75	32.38	12.42	35.34
	$RGB + RGB_t$	14.76	37.35	15.32	39.04
	$RGB + GGG_t$	**15.39**	37.80	**16.97**	40.49
	$RGB + RRR_t$	14.51	38.14	15.39	40.09
	$RGB + BBB_t$	14.16	34.91	14.53	36.72

poral features. Therefore, the fusion of the outputs of the two streams can represent the spatiotemporal features of the entire video. We explore different combinations of RGB and X_t frames. The results of our experiments are shown in Table 6.1. For all combinations and for both datasets, the performances are better than baseline (a single 2D-CNN operating on standard RGB frames). All the improvements of network performances suggest that the proposed X_t frames provides useful temporal information for temporal stream in addition to the spatial information captured by the spatial stream. Note that the fusion of RGB and GGG_t obtains top-1 video accuracy improvements of 2.43% and 4.55% on UCF101 and HMDB51 respectively, which are the best results for both datasets.

Evaluation with 3D-CNNs. 3D kernels can extract spatiotemporal features from plain RGB frames but only from a small neighborhood. Thus, 3D-CNNs have limitations in modeling long-range temporal dependencies due to limited receptive fields. Our X_t frames provide three times longer temporal sequences than that provided by plain RGB frames, enabling the 3D-CNN to capture richer temporal information. To explore the impact of different color information, we evaluate the network performances on different X_t frames and compare them with network performances on plain RGB frames. Table 6.2 shows the results we obtained. Similar to the results on 2D-CNN, all network performances obtained on X_t frames are better than those on plain RGB frames. GGG_t frames achieve the best top1 accuracy of 55.07% and 28.08% on UCF101 and HMDB51 datasets respectively, which are 8.32% and 3.56% higher than the RGB baseline on UCF101 and HMDB51 datasets.

Evaluation with Transformers. We also conduct experiments and compare results on RGB and X_t frames for Transformers. All the results are shown in Table 6.3. We find

Table 6.2 Results on 3D-ResNet18

Dataset	Modality	Top1	Top5
UCF101	RGB	46.75	76.61
	RGB_t	50.79	78.36
	$\underline{GGG_t}$	**55.07**	80.71
	$\overline{RRR_t}$	54.73	80.23
	BBB_t	53.22	79.12
HMDB51	RGB	24.52	52.54
	RGB_t	26.17	53.53
	$\underline{GGG_t}$	**28.08**	57.35
	$\overline{RRR_t}$	26.96	56.30
	BBB_t	26.37	55.37

Table 6.3 Results on UniFormerV2

Dataset	Modality	Top1	Top5
UCF101	RGB	45.38	72.23
	RGB_t	47.83	75.24
	$\underline{GGG_t}$	**57.82**	82.95
	$\overline{RRR_t}$	57.32	82.45
	BBB_t	55.79	81.69
HMDB51	RGB	23.47	55.44
	RGB_t	26.57	56.89
	$\underline{GGG_t}$	**35.40**	68.30
	$\overline{RRR_t}$	33.42	66.45
	BBB_t	31.58	62.69

that the performance of RGB frames on both datasets for Transformers are worse than with 3D-CNNs, because UCF101 and HMDB51 are small video datasets and the backbone UniFormerV2 is much deeper than 3D-ResNet18. Transformers are proven to be better at capturing global information, but lack some of the inductive biases compared to CNNs and can thus not generalize well when trained on small datasets (Dosovitskiy et al. 2020*b*). However, all performances on our X_t frames with the same color outperform the performances on 3D-ResNet18, which suggests that our X_t frames help the Transformers generalize better on small datasets. And again, the performances on GGG_t frames achieve the highest top1 accuracy on both dataset, obtaining 12.44% and 11.93% improvements on UCF101 and HMDB51 datasets respectively.

By analyzing the results shown in Table 6.2 and Table 6.3, we can draw some

Table 6.4 Fusion on 3D-ResNet18

Dataset	Modality	Top1	Top5
UCF101	RGB	46.75	76.61
	$RGB + RGB_t$	50.90	79.57
	$RGB + GGG_t$	**57.03**	81.87
	$RGB + RRR_t$	57.00	81.47
	$RGB + BBB_t$	56.21	82.19
HMDB51	RGB	24.52	52.54
	$RGB + RGB_t$	26.70	55.17
	$RGB + GGG_t$	**31.64**	58.14
	$RGB + RRR_t$	30.59	58.21
	$RGB + BBB_t$	30.39	58.27

conclusions. The performance improvements obtained by RGB_t frames suggests that our color-sampling strategy enables the networks to model longer range temporal sequences. And the further performance gains obtained by GGG_t frames indicates that the green channel contains richer spatial information than other color channels. The BBB_t frame obtains lower accuracy than the others, which may be due to the blue channel being noisier because of higher amplification.

Fusion. Finally, we explore the fusion of spatial and temporal streams on 3D-CNNs and Transformers to further improve the performance. The fusion results of different modalities are shown in Table 6.4 and Table 6.5. The combination of RGB and GGG_t obtains a 10.28% accuracy gain on UCF101 dataset and a 7.12% accuracy gain on HMDB51 dataset for 3D-CNN, it also achieves performance improvements of 15.11% and 13.71% on the UCF101 and HMDB51 datasets respectively for Transformer. Note that all results obtained by a two-stream network are better than those for a single-stream network.

The overview of all results obtained with 3D-CNNs and Transformers are shown in Fig. 6.6 and Fig. 6.7. The performance curves illustrate an ordering of how much the different color-sampling strategies improve the results: $GGG_t > RRR_t > BBB_t > RGB_t$. A single X_t frame fuses temporal differences sampled from three different RGB frames, so that a video clip consisting of X_t frames can represent three times longer temporal information than clip formed using standard RGB frames. That is the reason why X_t frames improve the performances for both dataset. And all X_t frames with the same color, such as GGG_t, achieve higher performances than RGB_t frames; the reason may be that in RGB_t frames, color differences and time differences are coupled together, so that the channel dimension of 3D kernels has to capture color and temporal feature at the same time. We also find the GGG_t frames obtain better network

Table 6.5 Fusion on UniFormerV2

Dataset	Modality	Top1	Top5
UCF101	RGB	45.38	72.23
	$RGB + RGB_t$	49.66	75.82
	$RGB + GGG_t$	**60.49**	83.96
	$RGB + RRR_t$	60.10	83.93
	$RGB + BBB_t$	59.38	83.11
HMDB51	RGB	23.47	55.44
	$RGB + RGB_t$	26.83	58.87
	$RGB + GGG_t$	**37.18**	68.36
	$RGB + RRR_t$	36.12	66.84
	$RGB + BBB_t$	34.28	64.67

Table 6.6 Parameters, FLOPs and views for inference

Backbone	Parameters (M)	FLOPS (G)	Clips x Crops
ResNet18	11.23	1.82	1 x 1
3D-ResNet18	33.26	17.06	10 x 3
UniFormerV2	123.82	157.41	10 x 3

performance than RRR_t and BBB_t frames, which verifies our hypothesis that green channels contain more information than the other color channels. Moreover, all fusions of X_t frames and RGB frames lead to further performance gains. However, the fusion of spatial stream operating on RGB frames and temporal stream operating on plain RGB_t frames only improve the accuracy a bit, indicating that there is some redundancy in the color channels.

Table 6.6 illustrates the complexity of our networks. The number of parameters and FLOPs in the table demonstrates that the proposed strategies do not increase the computational costs. Our color sampling strategy improves performances by enabling the networks to model longer temporal dependencies without changing the input dimension. The last column of the table shows the number of temporal clips and spatial crops we use for the inference stage.

6.4 Conclusion

In this Chapter, we propose a novel color-sampling strategy that can help networks to model longer temporal sequences without increasing the input dimension. All the results we obtain demonstrate that trading color information for temporal information

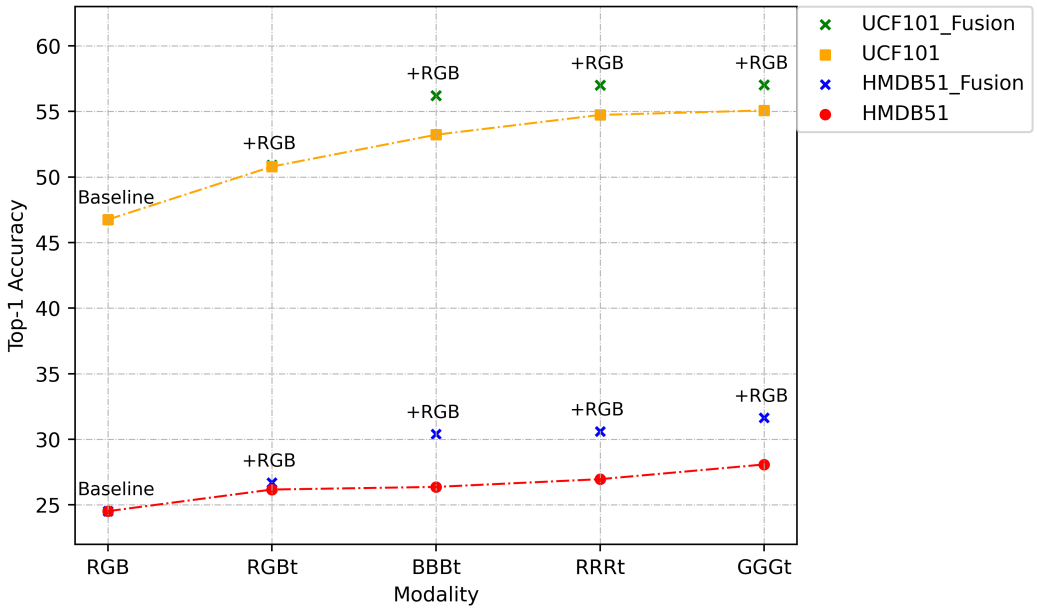


Fig. 6.6 Overview of results obtained with the 3D-ResNet18. Curves indicate top-1 accuracies obtained for the different sampling strategies and crosses the gain in accuracy when fusing two networks.

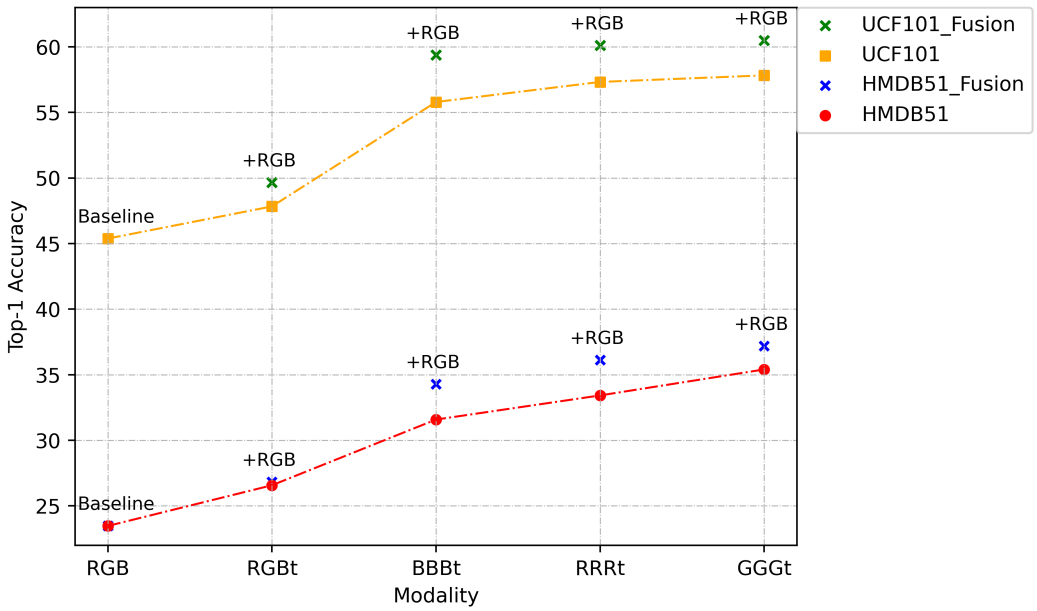


Fig. 6.7 Overview of results obtained with the UniFormerV2. Curves indicate top-1 accuracies obtained for the different sampling strategies and crosses the gain in accuracy when fusing two networks.

pays off in video understanding. Furthermore, we also verify that the two-stream network brings more improvements to network performance by fusing spatial stream operating on standard RGB frames with temporal stream operating on our proposed X_t frames, which contain richer temporal information. We find that among the three color channels, the green channel is the most informative for video understanding, thus obtaining the highest top1 accuracy among the three network architectures for both datasets.

To illustrate the effectiveness of our color sampling strategy, we train all networks that use ResNet18, 3D-ResNet18 and UniFormerV2 as backbones from scratch. On the one hand, all current state-of-the-art networks are first pretrained on large image datasets such as ImageNet and CLIP-400 and then further post-pretrained on huge video datasets such as Kinetics and Sport1M, but we can not afford such high computational costs. On the other hand, we propose a novel color sampling strategy to obtain X_t frames which are different from traditional RGB frames, so we could not use pretrained networks for transfer learning. We believe, however, that the demonstrated performance improvements of our network will transfer to large datasets.

CHAPTER 7

Machine Learning Model for Structural MRI Image Analysis

7.1 Introduction

Psychosis refers to a disturbance in the perception of reality, making it difficult to determine what is real and what is unreal. Psychosis involves some typical symptoms such as hallucinations, delusions and disordered thinking, speaking and behavior in certain conditions (Arciniegas 2015). Psychosis can be caused by psychiatric illnesses such as schizophrenia, major depression, bipolar disorder and Alzheimer's disease and also can be triggered by trauma, stress, head injury, and drug or alcohol misuse etc (Griswold et al. 2015, Davies 2017).

Psychosis may cause complications and even lead to self-harm or suicide. In addition, it reduces the quality of life of patients and their families as well as increases the burden on the healthcare system (James et al. 2018). Therefore, early diagnosis and treatment of psychosis is very necessary. Currently, the diagnostic criteria for psychosis are not strictly defined. The diagnosis of psychosis is primarily made based on possible causes and various clinical symptoms, including delusions or formal thought disorders. There are many studies showing structural changes in the brain as psychosis develops. However, medical images are currently not part of diagnosing psychosis and have no contribution to the treatment of psychosis (Sun et al. 2009, de Castro-Manglano et al. 2011, Ziermans et al. 2012, Andreou and Borgwardt 2020).

Nowadays, the goal of psychiatric research is not only to treat psychosis but also to predict it, allowing early intervention to delay the development of psychosis or even prevent its onset. Usually, the ultra-high risk criteria or the presence of basic symptoms such as changes in perception, speaking and behavior are used to determine clinical high risk of developing psychosis (CHR) (Andreou et al. 2023, 2019). Only 20% to 36% (Fusar-Poli et al. 2020, 2012) of subjects determined as CHR based on the above criteria develop psychosis. Therefore, we need more accurate criteria to predict whether CHR will transit into psychosis. Inspired by the theory of structural changes in brain, we can combine clinical criteria with medical image analysis to provide patients with better psychosis diagnosis and treatment.

To understand the mechanisms of psychotic disorders and possible indicators for di-

agnosing and predicting it, factors of the environmental, clinical and biological aspects have been investigated (Montemagni et al. 2020). There are related studies on structural MRI images of different measures including global volume (Montemagni et al. 2020, Koutsouleris et al. 2015, 2012, 2009), GM (Howes et al. 2023) and CSF volume (Dabiri et al. 2022), cortical thickness (Howes et al. 2023) and fractal dimension (FD) (Squarcina et al. 2015, Zhao et al. 2016, Yotter et al. 2011, Nenadic et al. 2014) in order to make the transition from CHR to psychosis predictable. These studies found reductions in fractal dimension, cortical thickness and volume of various regions.

Our work aims to design a model that can analyze psychosis by using the calculated fractal dimension of brain MRI images. The results show that fractal dimension is useful for distinguishing different groups from each other. Moreover, the fractal dimension appears to be a key indicator of whether a clinical high risk (CHR) patient will develop psychosis.

7.2 Methodology

7.2.1 Image Processing

We used structural MRI images ($121 \times 145 \times 121$) from a total of 194 participants. Firstly, neuroradiologists visually assess each MRI image and manually exclude visible abnormalities and artifacts. The MRI images were then preprocessed using the Cat12 toolbox (Gaser et al. 2024) from the Spm12 software package. All MRI images were segmented into gray matter (GM), white matter (WM), and cerebrospinal fluid (CSF). We use GM based MRI images as our segmented MRI to further process and analyse. Detailed information of MRI images and preprocessing procedure can be found in (Koutsouleris et al. 2015, 2009).

As Korda et al (Korda et al. 2022) described, voxel-by-voxel sliding 3D cubes are used on the segmented MRI images to calculate local values. The resulting overlapping cubes fill the entire brain. After experimental comparisons of different cube sizes, we selected the dimensions of 15x15x15 and 25x25x25. Note that only cubes that intersect with the brain region are considered, i.e. cubes containing only zero values are excluded. For each cube, we binarize them individually and then calculate the fractal dimension. The algorithm assigns the calculated fractal dimension value to the center of the cube.

To binarize the cube, we first chose a suitable threshold and then used Matlab to set all voxels with gray values less than the threshold to 0 and all voxels with gray values greater than the threshold to 1. The thresholds we used are 0.05, 0.3, 0.5 and 0.7. The fractal dimension is calculated individually for each binarized cube at different threshold.

7.2.2 Fractal Dimension

Fractal dimension is an index used to describe the complexity of an object's shape. It is used in medicine to describe the complex morphology of brain structures (such as cortex, gray matter, and white matter), thereby being an indicator for neuroimaging techniques such as MRI to distinguish different physiological and pathological conditions (Zhang et al. 2016). There have been some findings on how the fractal dimension of the brain changes with the development of certain neurological diseases. For example, FD has been shown to decrease with progressive degeneration in Alzheimer's disease and to increase with the development of tumors or epilepsy (John et al. 2015).

Mathematically, the fractal dimension quantifies the complexity as a ratio of the change in detail (N) to the change in scale (S) (Zhang et al. 2016). There are many ways to calculate FD, but they are all based on the same principle. We calculated the FD following the work of Beeyanal et al (Beyenal et al. 2004), a modified version of the Minkowski sausage method (Russ 2006). In 3D fractal dimension calculation, volume is swept out with increasing dilation of a sphere. If a sphere sweeps out the boundary, dilation is used to smooth the boundary line. Boundary pixels are set to zero and all other pixels are set to 1. We then use Matlab to calculate the Euclidean distance in three dimensions from each pixel in the cube to the boundary. To calculate the dilation volume, the sphere radius varies and the number of pixels at a distance smaller than this radius value in the cube are counted. Fractality is the slope of the straight line obtained by plotting ln(sphere radius) against ln(volume/radius ratio). The fractal dimension is defined as FD = 2- slope. Detailed description of the FD calculation can be found in (Beyenal et al. 2004).

7.2.3 Machine Learning

The calculated fractal dimensions are saved in the Neuroimaging Informatics Technology Initiative (NIfTI) format like segmented MRI images. The dimension of the data is still 3D: $121 \times 145 \times 121$. The 3D local fractal dimension calculated cube by cube contains more information than the global fractal dimension, which only uses a single value to quantify the complexity of the whole brain. However, the high dimensionality of local FD increases the difficulty of analysis. Therefore, it is necessary to find a method to reduce the dimensionality or summarize the local fractal dimension of the whole brain. Since statistical features have proven to be useful features for machine learning models, we use statistical features to summarize the local FD of the whole brain.

The local FD index quantifies the complexity of the cerebral cortex within each cube we use. Since each cube has a different local FD value, we can change our view of them and treat these values as the intensity of each voxel (Because the cubes are overlapping, each voxel has a local FD value.). Thus, we process fractal dimension features just

like medical images. Since we do not know which statistical features are useful for analyzing psychosis, we decided to use the Pyradiomics package (Van Griethuysen et al. 2017) to automatically extract a large variety of statistical features instead of manually extracting a few features. After that, automatic feature selection is performed. Our algorithm flow is shown in Figure 7.1.

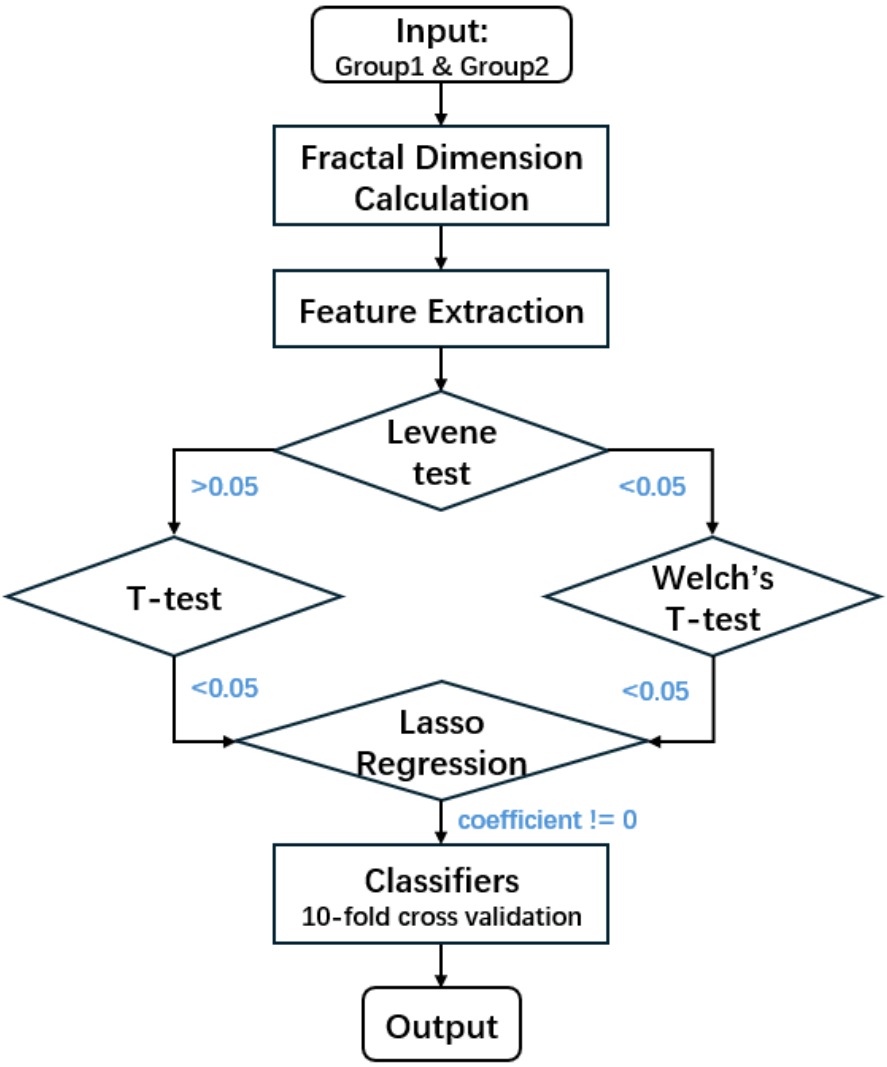


Fig. 7.1 The detailed flowchart of our algorithm (Group1 or Group2 is one of FEP, CHR_T, CHR_NT and HC).

7.2.3.1 Feature Extraction

We first mask the entire brain region because masks for different brain regions are not provided. We then use the Pyradiomics package to extract features from the area covered by the mask we created. These extracted features contain approximately 1500 features per image on average, with eight classes: First Order Statistics, Shape-based (2D), Shape-based (3D), Gray Level Co-occurrence Matrix (GLCM), Gray Level Dependence Matrix (GLDM), Gray Level Size Zone Matrix (GLSZM), Neighbouring Gray Tone

Difference Matrix (NGTDM) and Gray Level Run Length Matrix (GLRLM). All extracted features are written to comma separated values (.csv) files for saving. Although the dimension of extracted features is much lower than that of FD features, which is 2122945 (=121x145x121), there are still many redundant features that need to be excluded.

7.2.3.2 Feature Selection

Irrelevant features introduce redundant information, increase computational complexity and even hurt model performance, thus feature selection is an important procedure in machine learning, especially when dealing with high-dimensional data. A good feature selection method selects the most informative features, thereby speeding up model training and reasoning and improving model performance.

The features extracted by the Pyradiomics package also include string features, which need to be excluded first. We then randomly shuffle all retained features and use multiple feature selectors to select the most discriminative features for better classification.

We choose the classic T-test as our first feature selector. Before the T-test, the Levene's test (Levene et al. 1960) is used to check the homogeneity of variances for each feature in pairwise groups. Levene test is used to test the equality of variances between groups, and its results determine which kind of T-test methods to use next. There are two hypothesises for Levene test, H_0: the variance between groups is equal or H_1: at least one pair the variance is not equal to the others. The Levene test is defined as follows:

$$W = \frac{(N-k)}{(k-1)} \frac{\sum_{i=1}^{k} N_i (\overline{Z}_{i.} - \overline{Z}_{..})^2}{\sum_{i=1}^{k} \sum_{j=1}^{N_i} (Z_{ij} - \overline{Z}_{i.})^2}, \tag{7.1}$$

where N is the sample size of variable Y, N_i is the sample size of the i^{th} subgroup, k is the number of subgroups; $Z_{ij} = |Y_{ij} - \widetilde{Y}_{i.}|$, $\widetilde{Y}_{i.}$ is the median of the i^{th} subgroup. For most underlying distributions of the data, the median can provide good robustness and good performance (Brown and Forsythe 1974), so we choose the median instead of the mean and the trimmed mean; and $\overline{Z}_{i.}$ are the group means of the Z_{ij} and $\overline{Z}_{..}$ is the overall mean of the Z_{ij}.

The resulting p-value then indicates which hypothesis our feature fits. Empirically, we use a threshold of 0.05 to compare with the p-value. A p-value greater than the threshold satisfies the hypothesis H_0, suggesting that the variance of features between groups is equal; a p-value less than the threshold, contrary to the hypothesis H_0 but satisfies H_1, indicating that the variances of features between groups are unequal.

If the result of the Leneve test shows that the hypothesis H_0 is true, we should

perform a standard independent 2 sample T-test (Student 1908) with equal population variance assumption. Otherwise, we should perform Welch's T-test (Welch 1947) that assumes unequal group variances.

The standard independent 2-sample T-test is defined as follows:

$$t = \frac{\overline{X}_1 - \overline{X}_2}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \tag{7.2}$$

$$s_p = \sqrt{\frac{(n_1 - 1)s_{X_1}^2 + (n_2 - 1)s_{X_2}^2}{n_1 + n_2 - 2}},$$

where $\overline{X}_i$ is the mean value of sample set i and n_i is sample size, $s_{X_i}^2$ is the unbiased estimator of the population variance.

The definition of Welch's T-test is as follows:

$$t = \frac{\overline{X}_1 - \overline{X}_2}{s_{\overline{\triangle}}}, \tag{7.3}$$

$$s_{\overline{\triangle}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}},$$

where s_i is the corrected sample standard deviation.

We then compare the p-value resulted from the T-test with the threshold 0.05 to select the most informative features. For both T-test methods, a resulting p-value is less than 0.05 means that there is a statistically significant difference in the features between the compared two groups. Therefore, we should retain features with a T-test result p-value less than 0.05 and exclude features with a p-value greater than 0.05.

After the feature selection by T-test, we further use the Least Absolute Shrinkage and Selection Operator (Lasso) regression to select the most distinguishable features. Lasso regression is a regularization technique that applies penalties to improve model performance and prevent overfitting. It is commonly used for processing high-dimensional data and as a feature selector. By shrinking the coefficient values associated with less relevant predictors to zero, Lasso regression can reduce model complexity and improve classification performance by selecting informative features.

To implement Lasso regression, we preprocess the features selected by the T-test by first concatenating the two sets of features from different groups and then normalizing them by mean and standard deviation normalization. After that, we train a Lasso Regression model on the preprocessed features to conduct further feature selection. The cost function of Lasso regression is defined as follows:

$$J(w) = \min_w \frac{1}{2 * n} ||y - Xw||_2^2 + \alpha * ||w||_1, \tag{7.4}$$

where n is the number of training samples, y is the target value and X are the training data. w is the coefficient associated with predictor variable and α represents the regularization parameter that controls the strength of the L1 penalty.

We retain the most relevant features by excluding features whose coefficient value of Lasso regression is equal to zero. After multiple feature selections, we exclude highly irrelevant and redundant information, which helps speed up the training process and improve model performance.

7.2.3.3 Classification

After the feature selection procedure, we use machine learning models to make decision based on the selected features. Ten different classifiers are used for our binary classification tasks, they are Ada Boosting, Decision Tree, Random Forest, Gradient Boosting, Gaussian Naive Bayes (GaussianNB), K-Nearest Neighbors (KNN), Logistic Regression (LR), Multi-layer Perceptron (MLP), Stochastic Gradient Descent (SGD) and Support Vector Classifier (SVC). By comparing the results of different experimental groups on different data (segmented MRI and various FD features), we find that the performances of Logistic Regression (LR), Multi-layer Perceptron (MLP), Gaussian Naive Bayes (GaussianNB) and Support Vector Classifier (SVC) are more robust. Other classifiers performed less consistently, indicating that they may not be suitable for our data. It is possible that each classifier will work well with some data and not so well with others (Fernández-Delgado et al. 2014), so we only report the results of the remaining classifiers.

Logistic regression is a generalized linear model, mainly used for binary classification tasks. It maps the output of the linear regression model through a logistic function and limits the output to the range of 0 and 1 to represent the probability that an instance belongs to a certain class. The Multilayer Perceptron (MLP) consists of three types of fully connected layers. The input layer receives input data, the hidden layer learns feature representations, and the output layer generates predictions. The MLP introduces nonlinearities by using activation functions, enabling the network to learn complex patterns from the data. Gaussian Naive Bayes (GaussianNB) hypothesizes that each feature follows a Gaussian distribution and features for a given class label are independent. It calculates the probability that a sample belongs to a certain class based on Bayes' theorem. Support Vector Classifier (SVC) aims to find the optimal hyperplane in high-dimensional space that separates data points of different classes in the feature space. The optimal hyperplane is the one with the maximum margin between the closest points of different classes. For all the classifiers, an input is classified as a positive sample if the probability is greater than 0.5, otherwise it is a negative sample.

Table 7.1 Overview of the dataset

Class	#Participant
first-episode psychosis (FEP)	77
clinic high risk with transition (CHR_T)	16
clinic high risk without transition (CHR_NT)	57
healthy control (HC)	44

7.2.3.4 Performance Metrics

Empirically, we choose balanced accuracy, sensitivity, and specificity as performance metrics as in other medical studies. We use balanced accuracy rather than accuracy because our dataset is highly imbalanced. However, balanced accuracy does not provide information about how many samples of each group are correctly classified, which is very important for medical research. So we also use sensitivity and specificity to indicate the ratio of positive and negative samples being correctly classified, respectively.

$$Balanced\ Accuracy = \frac{Specificity + Sensitivity}{2}, \tag{7.5}$$

where the formulas of Specificity and Sensitivity can be found in Chapter 5.

7.3 Experiments and Discussions

7.3.1 Dataset

The dataset is a part of the Early Detection of Psychosis project (FePsy) at the Department of Psychiatry, University of Basel, Switzerland (Riecher-Rössler et al. 2007). The dataset contains four different groups as shown in Table 7.1. Specifically, there are 77 first-episode psychosis (FEP) participants, 16 clinic high risk with transition (CHR_T) participants, 57 clinic high risk without transition (CHR_NT) participants and 44 healthy control (HC) participants. For each participant, there are a segmented MRI image (as shown in Fig. 7.2) and FD features calculated according to various conditions based on the MRI image: the combinations of cube_size 15 or 25 and thresholds 0.05, 0.3, 0.5 or 0.7.

7.3.2 Implementation Details

The Pyradiomic library (Van Griethuysen et al. 2017) is used for feature extraction, both the SciPy library (Virtanen et al. 2020) and the Scikit-learn library (Pedregosa et al. 2011) are used for feature selection and the Scikit-learn library is used for Machine Learning classifiers. Since our dataset is small and unbalanced, we apply ten-fold cross-validation to evaluate model performance. For each fold, we randomly split the training set and the validation set in a ratio of 0.8:0.2. Moreover, the CHR_Ts group is used as

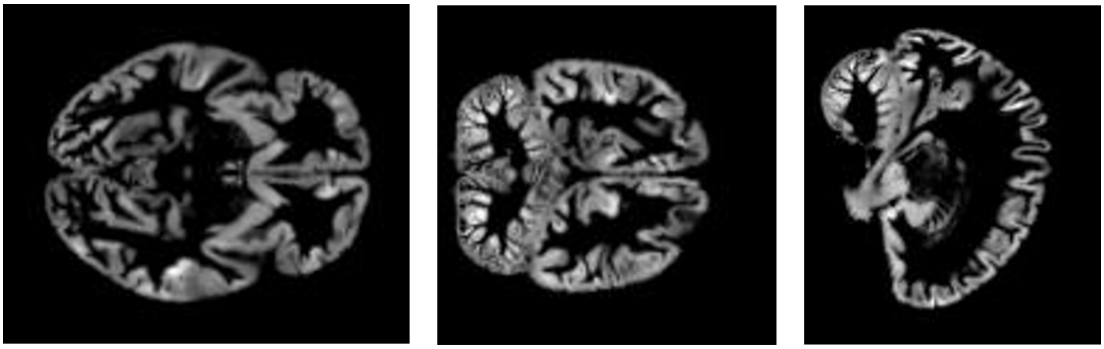


Fig. 7.2 Examples of segmented MRI images

an additional validation set for other models that do not include it, i.e. FEP vs. HC, FEP vs. CHR_NT and CHR_NT vs. HC. We conduct all experiments on a single GeForce RTX 3060 GPU.

7.3.3 Results and Discussions

7.3.3.1 Clinic High Risk with vs. without Transition

On the one hand, psychosis not only greatly affects the life quality of patients and their families, but also places a burden on the healthcare system. On the other hand, First-episode psychosis (FEP) usually does not appear suddenly but develops with gradual changes in the patient's perceptions and thoughts. Therefore, early intervention for psychosis is necessary and possible. So it is important to understand whether psychosis is predictable in order to identify potential psychosis and intervene early. However, the early signs of FEP may not be enough to alert patients and physicians. Therefore, we use MRI images and FD features to quantitatively analyze psychosis rather than these subtle signs and further aim to slow or even prevent clinic high-risk subjects (CHRs) from developing psychosis by introducing specialized treatments.

Hence, we first compare the CHR_T and CHR_NT groups to see if there are differences between them. If there are significant differences, then it is possible for doctors to screen for potential psychosis as early as possible and take preventive measures accordingly. The results are shown in Table 7.2.

For segmented MRI, we notice that the specificity of all classifiers are rather low, meaning that CHR_T could not be distinguished from CHR_NT based on the features extracted from segmented MRI. However, the specificity of all classifiers improved significantly based on the calculated fractal dimension (FD) with cube size 25 and thresholds 0.3 and 0.7, respectively, resulting in better model performance. All balanced accuracies (BAcs) are higher than 74.55%, and the highest one achieves 83.93%. Although our dataset is very imbalanced: 16 CHR_T and 57 CHR_NT, using FD features

Table 7.2 Comparison of clinic high risk with transition (CHR_T) and clinic high risk without transition (CHR_NT).

	Metrics	LR	MLP	GNB	SVC
Segmented	**Spe %**	31.25	43.75	56.25	31.25
	Sen %	89.47	85.96	84.21	87.72
	Bac %	60.36	64.86	70.23	59.48
FD25_30	**Spe %**	**75.00**	68.75	62.50	62.50
	Sen %	89.47	89.47	89.47	94.74
	Bac %	82.24	79.11	75.99	78.62
FD25_70	**Spe %**	56.25	68.75	**75.00**	56.25
	Sen %	92.86	91.07	92.86	**96.43**
	Bac %	74.55	79.91	**83.93**	76.34

* LR: Logistic Regression; MLP: Multi-layer Perceptron, GNB: Gausian Naive Bayes and SVC: Support Vector Classifier.
* Spe: specificity; Sen: sensitivity; Bac: balanced accuracy.

makes the performance comparable to other studies (Andreou et al. 2019, Fusar-Poli et al. 2015, 2020). All results on FD features suggest that whether CHRs transition to FEPs is predictable.

7.3.3.2 First-Episode Psychosis vs. Other Groups

First-episode psychosis (FEP) is when a patient starts to confuse reality with hallucinations or delusions. Early treatment for FEP can slow down, stop, and even reverse the progression of psychosis, thereby preserving the life quality of patients. Accurate and reliable early detection and diagnosis is vital for prompt treatment. To verify whether FD features can help early diagnosis of FEP, we conduct experiments to compare the following three pairwise groups: FEP vs. HC, FEP vs. CHR_NT and FEP vs. CHR_T.

Table 7.3 Comparison of first-episode psychosis (FEP) and healthy control (HC).

	Metrics	LR	MLP	GNB	SVC
Segmented	**Spe %**	77.92	76.62	75.32	74.03
	Sen %	56.82	52.27	54.55	47.73
	Bac %	67.37	64.45	64.94	60.88
FD15_05	**Spe %**	80.52	75.32	75.32	84.42
	Sen %	68.18	**72.73**	70.45	63.64
	Bac %	74.35	74.03	72.89	74.03
FD25_30	**Spe %**	85.71	81.82	71.43	**92.21**
	Sen %	63.64	65.91	63.64	63.64
	Bac %	74.68	73.87	67.54	**77.93**

First and most importantly, we have to prove that first-episode psychosis can be distinguished from healthy control. Thus, we compare FEP with HC. Table 7.3 shows the results.

From the results for segmented MRI images, we note that the sensitivity of all classifiers are lower than 56.82%, indicating that it is difficult for the classifiers to differentiate healthy controls (HCs) from FEPs. By using FD features, the sensitivity of all classifiers are improved to some extent. This leads to the highest BAc of 77.93%, demonstrating the ability of our model to distinguish FEP from HC.

Secondly, we have to show that first-episode psychosis (FEP) is different from clinic high risk without transition (CHR_NT). So we next compare FEP with CHR_NT. Table 7.4 shows the results we obtain.

Table 7.4 Comparison of first-episode psychosis (FEP) and clinic high risk without transition (CHR_NT).

	Metrics	**LR**	**MLP**	**GNB**	**SVC**
Segmented	**Spe %**	80.50	76.64	81.82	83.12
	Sen %	71.93	63.16	50.88	57.89
	Bac %	76.22	69.89	66.35	70.51
FD15_30	**Spe %**	83.12	84.42	84.42	83.12
	Sen %	78.95	70.18	68.42	70.18
	Bac %	81.04	77.30	76.42	76.65
FD25_30	**Spe %**	83.12	81.82	71.43	**88.31**
	Sen %	78.95	**82.46**	68.42	61.40
	Bac %	81.03	**82.14**	69.92	74.86

By observing the above table, the sensitivity of segmented MRI images shows that classifiers are not sensitive enough to discriminate CHR_NT from FEP expect for the Logistic Regression classifier. By using FD features, the sensitivity values of all classifiers are improved by 3.51% to 19.30%, resulting in significant improvements of balanced accuracies for all classifiers and obtaining the highest balanced accuracy of 82.14%. The results indicate that there are significant difference between FEP and CHR_NT.

Further, we have to make sure that first-episode psychosis (FEP) can be differentiate from clinic high risk with transition (CHR_T). We therefore compare FEP and CHR_T, and results are shown in Table 7.5.

From the results in Table 7.5, we find that the sensitivity values of classifiers are extremely low for segmented MRI images, meaning that classifiers are unable to distinguish CHR_T from FEP based on segmented MRI. FD features largely improves the sensitivity of all classifiers, resulting in most balanced accuracies exceeding 71.23%. The highest balanced accuracy of 79.83% is achieved for FD15_05, which indicates

Table 7.5 Comparison of first-episode psychosis (FEP) and clinic high risk with transition (CHR_T).

	Metrics	LR	MLP	GNB	SVC
Segmented	**Spe %**	97.40	94.81	93.51	96.10
	Sen %	12.50	25.00	43.75	6.25
	Bac %	54.95	59.90	68.63	51.18
FD15_05	**Spe %**	90.91	90.91	90.91	97.40
	Sen %	**68.75**	**68.75**	56.25	31.25
	Bac %	**79.83**	**79.83**	73.58	64.33
FD15_30	**Spe %**	96.10	93.51	97.40	**98.70**
	Sen %	62.50	62.50	56.25	43.75
	Bac %	79.30	78.01	76.83	71.23

that there are significant differences between FEP and CHR_T.

In summary, all sensitivities obtained based on the original segmented MRI images are poor. By using the calculated FD features, the sensitivity values of all classifiers are greatly improved in all experiments, especially in the experiment that compares FEP with CHR_T. Thus, we draw the conclusion that there are significant differences in the fractal dimension of FEP and HC, FEP and CHR_NT and FEP and CHR_T. In other words, the fractal dimension is a useful feature for reliable and accurate diagnosis of first-episode psychosis.

7.3.3.3 Clinic High Risk with Transition vs. Healthy Control

Clinic high risk with transition (CHR_T) is the stage preceding first-episode psychosis (FEP). Subjects in the CHR group are at high risk of developing psychosis, and CHR_T are the subjects who eventually developed psychosis. The above experiments prove that CHR_T can be distinguishable from FEP. Since CHR_T subjects eventually became FEPs, we hypothesize that there are significant differences between CHR_T and HC. We next compare CHR_T with HC to verify our hypothesis.

By analysing the results shown in Table 7.6, we note that the specificity values of different classifiers are quite poor for segmented MRI images, similar to the results of CHR_T vs. CHR_NT. And using FD features largely improved all specificity values. By using FD25_70 data, we achieve the highest balanced accuracy of 88.35%, which indicates that there are significant differences between CHR_T and HC.

7.3.3.4 Clinic High Risk without Transition vs. Healthy Control

Clinic high risks without transition (CHR_NT) are the subjects in the CHR group that do not transition to psychosis. We also compare CHR_NT and HC to find out if there

Table 7.6 Comparison of clinic high risk with transition (CHR_T) and healthy control (HC).

	Metrics	LR	MLP	GNB	SVC
Segmented	**Spe %**	43.75	43.75	43.75	37.50
	Sen %	88.64	86.36	86.36	90.91
	Bac %	66.19	65.06	65.06	64.20
FD25_30	**Spe %**	75.00	50.00	68.75	56.25
	Sen %	93.18	93.18	95.45	**97.73**
	Bac %	84.09	71.59	82.10	76.99
FD25_70	**Spe %**	75.00	68.75	**81.25**	75.00
	Sen %	90.91	88.64	95.45	93.18
	Bac %	82.95	78.69	**88.35**	84.09

are also large differences between these two groups. If so, doctors should conduct continual disease monitoring to prevent the progression of psychosis. Results are shown in Table 7.7.

Table 7.7 Comparison of clinic high risk without transition (CHR_NT) and healthy control (HC).

	Metrics	LR	MLP	GNB	SVC
Segmented	**Spe %**	77.19	71.93	80.70	77.19
	Sen %	59.09	54.55	63.64	52.27
	Bac %	68.14	63.24	72.17	64.73
FD15_50	**Spe %**	80.70	71.93	75.44	80.70
	Sen %	65.91	72.73	61.36	56.82
	Bac %	73.31	72.33	68.40	68.76
FD15_70	**Spe %**	**85.96**	**85.96**	82.46	77.19
	Sen %	**77.27**	68.18	45.45	65.91
	Bac %	**81.62**	77.07	63.96	71.55

From the above results, we notice that the sensitivity values for segmented MRI images of all classifiers are between 52.27% and 63.64%, which means that it is difficult for the classifiers to distinguish HC from CHR_NT. Similar to other experimental results, FD features improve the sensitivity of most classifiers to some extent, except for the GaussianNB, thereby achieving the highest balanced accuracy of 81.62%. The performance gain brought by FD features suggests that CHR_NT can be differentiated from HC.

7.3.3.5 Using Clinic High Risk with Transition as Additional Validation

We use ten-fold cross-validation for the above experiments, then CHR_Ts are used as an additional test set for the experiments without CHR_T, i.e. FEP vs. HC, FEP vs. CHR_NT and CHR_NT vs. HC. We first extract from the CHR_T group the same features selected by the feature selector for each model, and then feed these features into the corresponding test model to make predictions. The results for each model are as follows:

Table 7.8 Clinic high risk with transition (CHR_T) as the test set.

	FEP vs. HC
Label	FEP: 0, HC: 1
Model	SVC with FD25_30
Predictions	[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
	CHR_NT vs. HC
Label	CHR_NT: 0, HC: 1
Model	LR with FD15_70
Predictions	[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
	FEP vs. CHR_NT
Label	FEP: 0, CHR_NT: 1
Model	MLP with FD25_30
Predictions	[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

By analysing the test results, we find that in the comparison of i) FEP and HC, all CHR_Ts are predicted as FEPs, which suggests that CHR_T is more similar to FEP relative to HC; ii) CHR_T and HC, all CHR_Ts are predicted as CHR_NTs, which indicates that CHR_T is more like CHR_NT than HC; iii) FEP and CHR_T, all CHR_Ts are predicted as CHR_NTs, which indicates that CHR_T is closer to CHR_NT than FEP.

7.4 Conclusion

In this Chapter, we design a machine learning model to classify different groups related to psychosis based on calculated FD features. We conducted experiments on four groups in pairs, i.e. CHR_T vs. CHR_NT, FEP vs. HC, FEP vs. CHR_NT, FEP vs. CHR_T, CHR_T vs. HC and CHR_NT vs. HC, by using different fractal dimensions with various cube sizes (15 and 25) and thresholds (0.05, 0.3, 0.5 and 0.7). Compared to the results obtained by segmented MRI, all the results on fractal dimensions show that fractal dimension features can improve model performance, making these groups distinguishable, which demonstrates that the fractal dimension is a useful indicator for transition prediction in the CHR group and psychosis diagnosis. However, we do not find an optimal combination for fractal dimensions that suits any condition. The optimal

cube size and threshold need to be learned through practice. And from the additional experiment using CHR_T as test set, we find that CHR_T is more similar to CHR_NT whether compared to HC or compared to FEP. We think the reason is that CHR_T eventually transitioned to FEP, so it is different from HC, but has not yet developed to FEP so it also differs from FEP. Besides, both CHR_T and CHR_NT belong to the CHR group, so they are more similar. The effectiveness of FD features also reveals structural changes in the brain as psychosis progresses.

CHAPTER 8

Conclusions and Future Work

In Chapters 1 and 2, we introduce the topic of video-based action recognition, its importance, and its classic and SOTA architectures. In Chapter 3, 4 and 6, we propose different designs and architectures for video understanding and all of them achieve performance improvements. In Chapter 5 and 7, we present the application of machine learning methods in medicine. Specifically, we apply our architecture for video processing to glaucoma and visual impairment analysis and we design a machine learning model for psychosis analysis based on MRI data.

Our main contributions for video analysis are that 1) we introduce a RGB_t sampling strategy to capture longer temporal information without changing the input size and without increasing computational costs; 2) we design various-sized tubes for input tokenization to embed richer temporal information into tokens; 3) we introduce a bio-inspired and nonlinear connected MinBlock to select more informative features; 4) we introduce novel spatiotemporal slices to 'visualize' the motion trajectory and a saliency based sampling strategy to select the most useful slices, and design several 2D-CNN based architectures to evaluate the effectiveness of our spatiotemporal slices; 5) we explore a more effective use of color and temporal information, and we find that the trading of color for temporal information can improve the performance.

Our main contributions in medicine are that 1) we successfully apply our slicing 2D-CNN architectures to glaucoma diagnosis and visual impairment detection and find that there are relations between visual impairments and walking patterns; 2) we design a machine learning model for psychosis diagnosis and we obtain very good results in predicting whether clinic high-risk patients will transition into psychosis.

When exploring video understanding architectures, the biggest challenge we encounter was the limitation of computing resources. This forces us to evaluate our design on small datasets only. Our future plan is to adapt our design to existing pretrained backbones to bring network performance to the stat-of-the-art. And we would like to further apply our algorithm to medical applications, such as the diagnosis and monitoring of neurodegenerative disorders.

REFERENCES

Abdelbaky, A. and Aly, S. (2020), 'Human action recognition using short-time motion energy template images and pcanet features', *Neural Computing and Applications* **32**(16), 12561–12574.

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B. and Vijayanarasimhan, S. (2016), 'Youtube-8m: A large-scale video classification benchmark', *arXiv preprint arXiv:1609.08675* .

Aggarwal, J. K. and Ryoo, M. S. (2011), 'Human activity analysis: A review', *Acm Computing Surveys (Csur)* **43**(3), 1–43.

Andrearczyk, V. and Whelan, P. F. (2018), 'Convolutional neural network on three orthogonal planes for dynamic texture classification', *Pattern Recognition* **76**, 36–49.

Andreou, C., Bailey, B. and Borgwardt, S. (2019), 'Assessment and treatment of individuals at high risk for psychosis', *BJPsych Advances* **25**(3), 177–184.

Andreou, C. and Borgwardt, S. (2020), 'Structural and functional imaging markers for susceptibility to psychosis', *Molecular psychiatry* **25**(11), 2773–2785.

Andreou, C., Eickhoff, S., Heide, M., de Bock, R., Obleser, J. and Borgwardt, S. (2023), 'Predictors of transition in patients with clinical high risk for psychosis: an umbrella review', *Translational Psychiatry* **13**(1), 286.

Arciniegas, D. B. (2015), 'Psychosis', *CONTINUUM: lifelong learning in neurology* **21**(3), 715–736.

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', pp. 6836–6846.

Ba, J. L. (2016), 'Layer normalization', *arXiv preprint arXiv:1607.06450* .

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C. and Baskurt, A. (2011), Sequential deep learning for human action recognition, *in* 'Human Behavior Understanding: Second International Workshop, HBU 2011, Amsterdam, The Netherlands, November 16, 2011. Proceedings 2', Springer, pp. 29–39.

Barth, E. (2000*a*), 'A geometric view on early and middle level visual coding', *Spatial Vision* **13**(2–3), 193–199.

Barth, E. (2000*b*), The minors of the structure tensor, *in* G. Sommer, ed., 'Mustererkennung 2000', Springer, Berlin, pp. 221–228.

Barth, E. and Watson, A. B. (2000), 'A geometric framework for nonlinear visual coding', *Optics Express* **7**(4), 155–165.

Basha, S. S., Pulabaigari, V. and Mukherjee, S. (2022), 'An information-rich sampling technique over spatio-temporal cnn for classification of human actions in videos', *Multimedia Tools and Applications* **81**(28), 40431–40449.

Benitez-Garcia, G., Olivares-Mercado, J., Sanchez-Perez, G. and Yanai, K. (2021), Ipn hand: A video dataset and benchmark for real-time continuous hand gesture recognition, *in* '2020 25th international conference on pattern recognition (ICPR)', IEEE, pp. 4340–4347.

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

Beyenal, H., Donovan, C., Lewandowski, Z. and Harkin, G. (2004), 'Three-dimensional biofilm structure quantification', *Journal of microbiological methods* **59**(3), 395–413.

Beyer, R., Al-Nosairy, K. O., Freitag, C., Stolle, F. H., Behrens, M., Prabhakaran, G. T., Thieme, H., Schega, L. and Hoffmann, M. B. (2024), 'Treadmill-walking impairs visual function in early glaucoma and elderly controls', *Graefe's Archive for Clinical and Experimental Ophthalmology* .
URL: *https://link.springer.com/10.1007/s00417-024-06530-w*

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A. and Gould, S. (2016), Dynamic image networks for action recognition, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 3034–3042.

Blank, M., Gorelick, L., Shechtman, E., Irani, M. and Basri, R. (2005), Actions as space-time shapes, *in* 'The Tenth IEEE International Conference on Computer Vision (ICCV'05)', pp. 1395–1402.

Bobick, A. F. and Davis, J. W. (2001), 'The recognition of human movement using temporal templates', *IEEE Transactions on pattern analysis and machine intelligence* **23**(3), 257–267.

Brown, M. B. and Forsythe, A. B. (1974), 'Robust tests for the equality of variances', *Journal of the American statistical association* **69**(346), 364–367.

Bull, D. (2014), *Communicating pictures: A course in Image and Video Coding*, Academic Press.

Caba Heilbron, F., Escorcia, V., Ghanem, B. and Carlos Niebles, J. (2015), Activitynet: A large-scale video benchmark for human activity understanding, *in* 'Proceedings of the ieee conference on computer vision and pattern recognition', pp. 961–970.

Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C. and Zisserman, A. (2018), 'A short note about kinetics-600', *arXiv preprint arXiv:1808.01340* .

Carreira, J., Noland, E., Hillier, C. and Zisserman, A. (2019), 'A short note on the kinetics-700 human action dataset', *arXiv preprint arXiv:1907.06987* .

Carreira, J. and Zisserman, A. (2017), Quo vadis, action recognition? a new model and the kinetics dataset, *in* 'proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 6299–6308.

Dabiri, M., Dehghani Firouzabadi, F., Yang, K., Barker, P. B., Lee, R. R. and Yousem, D. M. (2022), 'Neuroimaging in schizophrenia: A review article', *Frontiers in neuroscience* **16**, 1042814.

Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W. et al. (2018), Scaling egocentric vision: The epic-kitchens dataset, *in* 'Proceedings of the European conference on computer vision (ECCV)', pp. 720–736.

Darafsh, S., Ghidary, S. S. and Zamani, M. S. (2021), 'Real-time activity recognition and intention recognition using a vision-based embedded system', *arXiv preprint arXiv:2107.12744* .

Davies, W. (2017), 'Understanding the pathophysiology of postpartum psychosis: Challenges and new approaches', *World journal of psychiatry* **7**(2), 77.

de Castro-Manglano, P., Mechelli, A., Soutullo, C., Gimenez-Amaya, J., Ortuño, F. and McGuire, P. (2011), 'Longitudinal changes in brain structure following the first episode of psychosis', *Psychiatry Research: Neuroimaging* **191**(3), 166–173.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. (2009), Imagenet: A large-scale hierarchical image database, *in* '2009 IEEE conference on computer vision and pattern recognition', Ieee, pp. 248–255.

Dollár, P., Rabaud, V., Cottrell, G. and Belongie, S. (2005), Behavior recognition via sparse spatio-temporal features, *in* '2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance', IEEE, pp. 65–72.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T. (2015), Long-term recurrent convolutional networks for visual recognition and description, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 2625–2634.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020*a*), 'An image is worth 16x16 words: Transformers for image recognition at scale', *arXiv preprint arXiv:2010.11929* .

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020*b*), 'An image is worth 16x16 words: Transformers for image recognition at scale', *arXiv preprint arXiv:2010.11929* .

Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J. and Feichtenhofer, C. (2021), Multiscale vision transformers, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 6824–6835.

Feichtenhofer, C. (2020), X3d: Expanding architectures for efficient video recognition, *in* 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition', pp. 203–213.

Feichtenhofer, C., Fan, H., Malik, J. and He, K. (2019), Slowfast networks for video recognition, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 6202–6211.

Fernández-Delgado, M., Cernadas, E., Barro, S. and Amorim, D. (2014), 'Do we need hundreds of classifiers to solve real world classification problems?', *The journal of machine learning research* **15**(1), 3133–3181.

Fusar-Poli, P., Bonoldi, I., Yung, A. R., Borgwardt, S., Kempton, M. J., Valmaggia, L., Barale, F., Caverzasi, E. and McGuire, P. (2012), 'Predicting psychosis: meta-analysis of transition outcomes in individuals at high clinical risk', *Archives of general psychiatry* **69**(3), 220–229.

Fusar-Poli, P., Cappucciati, M., Rutigliano, G., Schultze-Lutter, F., Bonoldi, I., Borgwardt, S., Riecher-Rössler, A., Addington, J., Perkins, D., Woods, S. W. et al. (2015), 'At risk or not at risk? a meta-analysis of the prognostic accuracy of psychometric interviews for psychosis prediction', *World Psychiatry* **14**(3), 322–332.

Fusar-Poli, P., De Pablo, G. S., Correll, C. U., Meyer-Lindenberg, A., Millan, M. J., Borgwardt, S., Galderisi, S., Bechdolf, A., Pfennig, A., Kessing, L. V. et al. (2020),

'Prevention of psychosis: advances in detection, prognosis, and intervention', *JAMA psychiatry* **77**(7), 755–765.

Gaser, C., Dahnke, R., Thompson, P. M., Kurth, F., Luders, E., Initiative, A. D. N. et al. (2024), 'Cat: a computational anatomy toolbox for the analysis of structural mri data', *GigaScience* **13**, giae049.

Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M. et al. (2017), The" something something" video database for learning and evaluating visual common sense, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 5842–5850.

Griswold, K. S., Del Regno, P. A. and Berger, R. C. (2015), 'Recognition and differential diagnosis of psychosis in primary care', *American family physician* **91**(12), 856–863.

Grüning, P. and Barth, E. (2022), 'Bio-inspired min-nets improve the performance and robustness of deep networks', *arXiv preprint arXiv:2201.02149* .

Grüning, P. and Barth, E. (2023), 'Efficient coding in human vision as a useful bias in computer vision and machine learning', *Journal of Perceptual Imaging* **6**, 1–10.

Grüning, P., Martinetz, T. and Barth, E. (2022), 'Fp-nets as novel deep networks inspired by vision', *Journal of Vision* **22**(1), 8–8.

Hara, K., Kataoka, H. and Satoh, Y. (2018), Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?, *in* 'Proceedings of the IEEE conference on Computer Vision and Pattern Recognition', pp. 6546–6555.

Hochreiter, S. and Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

Horn, B. K. and Schunck, B. G. (1981), 'Determining optical flow', *Artificial intelligence* **17**(1-3), 185–203.

Howes, O. D., Cummings, C., Chapman, G. E. and Shatalina, E. (2023), 'Neuroimaging in schizophrenia: an overview of findings and their implications for synaptic changes', *Neuropsychopharmacology* **48**(1), 151–167.

Hu, Y. and Barth, E. (2024), Video understanding using 2d-cnns on salient spatio-temporal slices, *in* 'International Conference on Artificial Neural Networks', Springer, pp. 256–270.

Ioffe, S. (2015), 'Batch normalization: Accelerating deep network training by reducing internal covariate shift', *arXiv preprint arXiv:1502.03167* .

Jähne, B. (1993), *Spatio-temporal image processing: theory and scientific applications*, Springer.

James, S. L., Abate, D., Abate, K. H., Abay, S. M., Abbafati, C., Abbasi, N., Abbastabar, H., Abd-Allah, F., Abdela, J., Abdelalim, A. et al. (2018), 'Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the global burden of disease study 2017', *The Lancet* **392**(10159), 1789–1858.

John, A. M., Elfanagely, O., Ayala, C. A., Cohen, M. and Prestigiacomo, C. J. (2015), 'The utility of fractal analysis in clinical neuroscience', *Reviews in the Neurosciences* **26**(6), 633–645.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. (2014), Large-scale video classification with convolutional neural networks, *in* 'Proceedings of the IEEE conference on Computer Vision and Pattern Recognition', pp. 1725–1732.

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* .

Kim, T.-K., Wong, S.-F. and Cipolla, R. (2007), Tensor canonical correlation analysis for action classification, *in* '2007 IEEE Conference on Computer Vision and Pattern Recognition', IEEE, pp. 1–8.

Klaser, A., Marszałek, M. and Schmid, C. (2008), A spatio-temporal descriptor based on 3d-gradients, *in* 'BMVC 2008-19th British Machine Vision Conference', British Machine Vision Association, pp. 275–1.

Korda, A., Luebeck, U., Andreou, C., Rogg, H. V., Avram, M., Ruef, A., Davatzikos, C., Koutsouleris, N. and Borgwardt, S. (2022), 'Identification of texture mri brain abnormalities on rst-episode psychosis and clinical high-risk patients using explainable artificial intelligence', *structure* **25**, 27.

Koutsouleris, N., Borgwardt, S., Meisenzahl, E. M., Bottlender, R., Möller, H.-J. and Riecher-Rössler, A. (2012), 'Disease prediction in the at-risk mental state for psychosis using neuroanatomical biomarkers: results from the fepsy study', *Schizophrenia bulletin* **38**(6), 1234–1246.

Koutsouleris, N., Meisenzahl, E. M., Davatzikos, C., Bottlender, R., Frodl, T., Scheuerecker, J., Schmitt, G., Zetzsche, T., Decker, P., Reiser, M. et al. (2009), 'Use of neuroanatomical pattern classification to identify subjects in at-risk mental states of psychosis and predict disease transition', *Archives of general psychiatry* **66**(7), 700–712.

Koutsouleris, N., Riecher-Rössler, A., Meisenzahl, E. M., Smieskova, R., Studerus, E., Kambeitz-Ilankovic, L., Von Saldern, S., Cabral, C., Reiser, M., Falkai, P. et al. (2015), 'Detecting the psychosis prodrome across high-risk populations using neuroanatomical biomarkers', *Schizophrenia bulletin* **41**(2), 471–482.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T. and Serre, T. (2011), Hmdb: a large video database for human motion recognition, *in* '2011 International conference on computer vision', IEEE, pp. 2556–2563.

Laptev, I. (2005), 'On space-time interest points', *International journal of computer vision* **64**, 107–123.

Levene, H. et al. (1960), 'Contributions to probability and statistics', *Essays in honor of Harold Hotelling* **278**, 292.

Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Wang, L. and Qiao, Y. (2022*a*), 'Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer', *arXiv preprint arXiv:2211.09552* .

Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Wang, L. and Qiao, Y. (2022*b*), 'Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer', *arXiv preprint arXiv:2211.09552* .

Lin, J., Gan, C. and Han, S. (2019), Tsm: Temporal shift module for efficient video understanding, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 7083–7093.

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', pp. 3202–3211.

Liu, Z., Wang, L., Wu, W., Qian, C. and Lu, T. (2021), Tam: Temporal adaptive module for video recognition, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 13708–13718.

Loshchilov, I. and Hutter, F. (2016), 'Sgdr: Stochastic gradient descent with warm restarts', *arXiv preprint arXiv:1608.03983* .

MacKay, D. G. and Ahmetzanov, M. V. (2005), 'Emotion, memory, and attention in the taboo stroop paradigm: An experimental analogue of flashbulb memories', *Psychological science* **16**(1), 25–32.

Maier-Hein, L., Mountney, P., Bartoli, A., Elhawary, H., Elson, D., Groch, A., Kolb, A., Rodrigues, M., Sorger, J., Speidel, S. and Stoyanov, D. (2013), 'Optical techniques for

3d surface reconstruction in computer-assisted laparoscopic surgery', *Medical Image Analysis* **17**(8), 974–996.

Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S. A., Yan, T., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C. et al. (2019), 'Moments in time dataset: one million videos for event understanding', *IEEE transactions on pattern analysis and machine intelligence* **42**(2), 502–508.

Montemagni, C., Bellino, S., Bracale, N., Bozzatello, P. and Rocca, P. (2020), 'Models predicting psychosis in patients with high clinical risk: a systematic review', *Frontiers in psychiatry* **11**, 223.

Neimark, D., Bar, O., Zohar, M. and Asselmann, D. (2021), Video transformer network, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 3163–3172.

Nenadic, I., Yotter, R. A., Sauer, H. and Gaser, C. (2014), 'Cortical surface complexity in frontal and temporal areas varies across subgroups of schizophrenia', *Human brain mapping* **35**(4), 1691–1699.

Neves, G. and Lagnado, L. (1999), 'The retina', *Current biology* **9**(18), R674–R677.

Pan, Y. (2010), 'Attentional capture by working memory contents.', *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale* **64**(2), 124.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. (2011), 'Scikit-learn: Machine learning in python', *the Journal of machine Learning research* **12**, 2825–2830.

Piergiovanni, A., Kuo, W. and Angelova, A. (2023), Rethinking video vits: Sparse video tubes for joint image and video learning, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 2214–2224.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al. (2021), Learning transferable visual models from natural language supervision, *in* 'International conference on machine learning', PMLR, pp. 8748–8763.

Riecher-Rössler, A., Gschwandtner, U., Aston, J., Borgwardt, S., Drewe, M., Fuhr, P., Pflüger, M., Radü, W., Schindler, C. and Stieglitz, R.-D. (2007), 'The basel early-detection-of-psychosis (fepsy)-study–design and preliminary results', *Acta Psychiatrica Scandinavica* **115**(2), 114–125.

Russ, J. C. (2006), *The image processing handbook*, CRC press.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.-C. (2018), Mobilenetv2: Inverted residuals and linear bottlenecks, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 4510–4520.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J. and Komatsuzaki, A. (2021), 'Laion-400m: Open dataset of clip-filtered 400 million image-text pairs', *arXiv preprint arXiv:2111.02114* .

Schuldt, C., Laptev, I. and Caputo, B. (2004), Recognizing human actions: a local svm approach, *in* 'Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.', Vol. 3, IEEE, pp. 32–36.

Scovanner, P., Ali, S. and Shah, M. (2007), A 3-dimensional sift descriptor and its application to action recognition, *in* 'Proceedings of the 15th ACM international conference on Multimedia', pp. 357–360.

Shen, X., Hua, G., Williams, L. and Wu, Y. (2012), 'Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields', *Image and Vision Computing* **30**(3), 227–235.

Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I. and Gupta, A. (2016), Hollywood in homes: Crowdsourcing data collection for activity understanding, *in* 'Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14', Springer, pp. 510–526.

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

Snowden, R. J., Snowden, R., Thompson, P. and Troscianko, T. (2012), *Basic vision: an introduction to visual perception*, Oxford University Press.

Soomro, K., Zamir, A. R. and Shah, M. (2012*a*), 'Ucf101: A dataset of 101 human actions classes from videos in the wild', *arXiv preprint arXiv:1212.0402* .

Soomro, K., Zamir, A. R. and Shah, M. (2012*b*), 'Ucf101: A dataset of 101 human actions classes from videos in the wild', *arXiv preprint arXiv:1212.0402* .

Squarcina, L., De Luca, A., Bellani, M., Brambilla, P., Turkheimer, F. E. and Bertoldo, A. (2015), 'Fractal analysis of mri data for the characterization of patients with schizophrenia and bipolar disorder', *Physics in Medicine and Biology* **60**(4), 1697.

Srivastava, S. and Sharma, G. (2024), Omnivec: Learning robust representations with cross modal sharing, *in* 'Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision', pp. 1236–1248.

Student (1908), 'The probable error of a mean', *Biometrika* pp. 1–25.

Sun, D., Phillips, L., Velakoulis, D., Yung, A., McGorry, P. D., Wood, S. J., van Erp, T. G., Thompson, P. M., Toga, A. W., Cannon, T. D. et al. (2009), 'Progressive brain structural changes mapped as psychosis develops in 'at risk'individuals', *Schizophrenia research* **108**(1-3), 85–92.

Tang, H., Liu, H., Xiao, W. and Sebe, N. (2019), 'Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion', *Neurocomputing* **331**, 424–433.

Tong, Z., Song, Y., Wang, J. and Wang, L. (2022), 'Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training', *Advances in neural information processing systems* **35**, 10078–10093.

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', pp. 4489–4497.

Tran, D., Wang, H., Torresani, L. and Feiszli, M. (2019), Video classification with channel-separated convolutional networks, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 5552–5561.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y. and Paluri, M. (2018), A closer look at spatiotemporal convolutions for action recognition, *in* 'Proceedings of the IEEE conference on Computer Vision and Pattern Recognition', pp. 6450–6459.

Van Griethuysen, J. J., Fedorov, A., Parmar, C., Hosny, A., Aucoin, N., Narayan, V., Beets-Tan, R. G., Fillion-Robin, J.-C., Pieper, S. and Aerts, H. J. (2017), 'Computational radiomics system to decode the radiographic phenotype', *Cancer research* **77**(21), e104–e107.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2023), 'Attention is all you need', *arXiv eprint arXiv:1706.03762* .

Verma, B. (2022), 'A two stream convolutional neural network with bi-directional gru model to classify dynamic hand gesture', *Journal of Visual Communication and Image Representation* **87**, 103554.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J. et al. (2020), 'Scipy 1.0: fundamental algorithms for scientific computing in python', *Nature methods* **17**(3), 261–272.

Wang, H. and Schmid, C. (2013), Action recognition with improved trajectories, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 3551–3558.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X. and Van Gool, L. (2016*a*), Temporal segment networks: Towards good practices for deep action recognition, *in* 'European conference on computer vision', Springer, pp. 20–36.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X. and Van Gool, L. (2016*b*), Temporal segment networks: Towards good practices for deep action recognition, *in* 'European conference on computer vision', Springer, pp. 20–36.

Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z. et al. (2022), 'Internvideo: General video foundation models via generative and discriminative learning', *arXiv preprint arXiv:2212.03191* .

Warren, D. H. and Strelow, E. R. (2013), *Electronic spatial sensing for the blind: contributions from perception, rehabilitation, and computer vision*, Vol. 99, Springer Science and Business Media.

Welch, B. L. (1947), 'The generalization of 'student's'problem when several different population varlances are involved', *Biometrika* **34**(1-2), 28–35.

Yan, S., Xiong, X., Arnab, A., Lu, Z., Zhang, M., Sun, C. and Schmid, C. (2022), Multiview transformers for video recognition, *in* 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition', pp. 3333–3343.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H. and Courville, A. (2015), Describing videos by exploiting temporal structure, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 4507–4515.

Yilmaz, A. and Shah, M. (2005), Actions sketch: A novel action representation, *in* '2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)', Vol. 1, IEEE, pp. 984–989.

Yotter, R. A., Nenadic, I., Ziegler, G., Thompson, P. M. and Gaser, C. (2011), 'Local cortical surface complexity maps from spherical harmonic reconstructions', *NeuroImage* **56**(3), 961–973.

Yu, J., Qin, M. and Zhou, S. (2022), 'Dynamic gesture recognition based on 2d convolutional neural network and feature fusion', *Scientific Reports* **12**(1), 4345.

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R. and Toderici, G. (2015), Beyond short snippets: Deep networks for video classification, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 4694–4702.

Zhang, L., Yue, G. and Ieva, A. (2016), 'The fractal geometry of the brain', *Springer series in computational neuroscience* .

Zhao, G., Denisova, K., Sehatpour, P., Long, J., Gui, W., Qiao, J., Javitt, D. C. and Wang, Z. (2016), 'Fractal dimension analysis of subcortical gray matter structures in schizophrenia', *PloS one* **11**(5), e0155415.

Zhu, Y., Li, X., Liu, C., Zolfaghari, M., Xiong, Y., Wu, C., Zhang, Z., Tighe, J., Manmatha, R. and Li, M. (2020), 'A comprehensive study of deep video action recognition', *arXiv preprint arXiv:2012.06567* .

Ziermans, T. B., Schothorst, P. F., Schnack, H. G., Koolschijn, P. C. M., Kahn, R. S., van Engeland, H. and Durston, S. (2012), 'Progressive structural brain changes during development of psychosis', *Schizophrenia bulletin* **38**(3), 519–530.

LIST OF PUBLICATIONS

1. Hu, Y., and Barth, E. (2024, June). Novel Design Ideas that Improve Video-Understanding Networks with Transformers. In 2024 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7). IEEE.

2. Guarischi, M., Hu, Y., Kurt, A. B., Zanchi, S., Barth, E., and Gori, M. (2024, June). A Machine Learning Approach to Unveil Balance Behavior Through Aging with an Auditory Cue. In 2024 IEEE International Symposium on Medical Measurements and Applications (MeMeA) (pp. 1-6). IEEE.

3. Hu, Y., and Barth, E. (2024, September). Video Understanding Using 2D-CNNs on Salient Spatio-Temporal Slices. In International Conference on Artificial Neural Networks (pp. 256-270). Cham: Springer Nature Switzerland.

4. Hu, Y., and Barth, E. (2025). How to Efficiently Use Color and Temporal Information for Video Understanding. In International Conference on Neural Information Processing (pp. 413-426). Springer, Singapore.

5. Hu, Y., Andac, S., Hoffmann, M. and Barth, E. A novel deep learning approach to assess visual system integrity from movement patterns during treadmill walking – a pilot study. (to be submitted)

6. Hu, Y., Frisman, M., Andreou, C., Avram, M., Riecher-Rössler, A., Borgwardt, S., ... and Korda, A. (2025). Brain Fractal Dimension and Machine Learning can predict first-episode psychosis and risk for transition to psychosis. Computers in Biology and Medicine, 193, 110333.

Appendix A

Major datasets for video understanding

Dataset	Year	#Video	#Class	Length
KTH (Schuldt et al. 2004)	2004	600	6	4s
Weizmann (Blank et al. 2005)	2005	90	10	3.66s
HMDB51 (Kuehne et al. 2011)	2011	6849	51	5s
UCF101 (Soomro et al. 2012*a*)	2012	13,320	101	6s
Sports-1M (Karpathy et al. 2014)	2014	1,133,158	487	5.5m
ActivityNet (Caba Heilbron et al. 2015)	2015	28,000	203	[5,10]m
YouTube-8M (Abu-El-Haija et al. 2016)	2016	8,000,000	4716	229.6s
Charades (Sigurdsson et al. 2016)	2016	9848	157	30.1s
Kinetics400 (Kay et al. 2017)	2017	306,245	400	10s
Kinetics600 (Carreira et al. 2018)	2018	495,547	600	10s
Kinetics700 (Carreira et al. 2019)	2019	650,317	700	10s
Something-Something V1 (Goyal et al. 2017)	2017	108.499	174	[2,6]s
Something-Something V2 (Goyal et al. 2017)	2017	220.847	174	[2,6]s
Moments in Time (Monfort et al. 2019)	2017	1,000,000	339	3s
EPIC-Kitchens (Damen et al. 2018)	2018	90,000	307	30s

Appendix B

The 32 selected classes of SthSth32

Class names	Index
Approaching [something] with your camera	0
Closing [something]	1
Folding [something]	2
Holding [something]	3
Holding [something] next to [something]	4
Moving [something] away from [something]	5
Moving [something] away from the camera	6
Moving [something] closer to [something]	7
Moving [something] down	8
Moving [something] towards the camera	9
Moving away from [something] with your camera	10
Opening [something]	11
Picking [something] up	12
Plugging [something] into [something]	13
Pretending to pick [something] up	14
Pretending to put [something] next to [something]	15
Pretending to put [something] on a surface	16
Pretending to take [something] from [somewhere]	17
Pushing [something] so that it slightly moves	18
Pushing [something] with [something]	19
Putting [something] into [something]	20
Showing a photo of [something] to the camera	21
Showing that [something] is empty	22
Stacking [number of] [something]	23
Throwing [something] against [something]	24
Turning [something] upside down	25
Turning the camera downwards while filming [something]	26
Turning the camera left while filming [something]	27
Turning the camera right while filming [something]	28
Turning the camera upwards while filming [something]	29
Uncovering [something]	30
Unfolding [something]	31

* Selected from the 40-selected classes reported in (Goyal et al. 2017), from Sth-Sth V2