ATTENTION MODELING WITH TEMPORAL SHIFT IN SIGN LANGUAGE RECOGNITION

by

Ahmet Faruk Çelimli

B.S., Computer Engineering, Boğaziçi University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2022

ACKNOWLEDGEMENTS

Firstly, I would like to thank my thesis advisor Prof. Lale Akarun for her expertise, patience and continuous guidance during my masters education. I would also like to thank Assist. Prof. Berk Gökberk and Prof. Hatice Köse for participating in my thesis jury and sharing their valuable comments and recommendations.

I am very thankful to Oğulcan Özdemir for his helpfulness, support and guidance on my thesis. I also would like to thank Boğaziçi University Perceptual Intelligence Laboratory members for their valuable friendship.

Finally, I would like to thank my family for their endless support, motivating words and kindness.

ABSTRACT

ATTENTION MODELING WITH TEMPORAL SHIFT IN SIGN LANGUAGE RECOGNITION

Sign languages (SLs) are the main communication language of deaf people. They are visual languages that establish communication through multiple cues including hand gestures, upper-body movements and facial expressions. Sign language recognition (SLR) models have the potential to ease communication between hearing and deaf people. Advancements in deep learning and the increased availability of public datasets have led more researchers to study SLR. These advancements shifted solution methods for SLR from hand-crafted features to 2 Dimensional Convolutional Neural Network (2D CNN) models. Inadequacy of 2D CNNs on temporal modeling and 3D CNNs' ability of spatio-temporal modeling made 3D CNNs a popular choice. Despite its successful results, high computational costs and memory requirements of 3D CNNs created a need for alternative architectures. In this thesis, we propose an SLR model that uses 2D CNN as backbone and attention modeling with temporal shift. Usage of 2D CNN decreases the number of parameters and required memory size compared to its 3D CNN counterpart. In order to increase adaptability to other datasets and simplify the training process our model uses full frame RGB images instead of cropped images that focus on specific body parts of signers. Since communication in SL is established by using multiple visual cues at the same time or at different moments, the model must learn how these cues are collaborating with each other. While temporal shift modules give our 2D CNN backbone model the ability of temporal modeling, attention modules learn to focus on what, where and when in videos. We tested our model with BosphorusSign22k dataset which is a Turkish isolated SLR dataset. The proposed model achieves 92.97% classification accuracy. Our study shows that attention modeling with temporal shift on top of 2D CNN backbone gives competitive results in isolated SLR.

ÖZET

İŞARET DİLİ TANIMADA ZAMANSAL KAYMA İLE DİKKAT MODELLEMESİ

İşaret dilleri, sağır bireylerin esas iletişim dilidir. Bu diller el şekilleri, üst vücut hareketleri ve yüz ifadeleri gibi birden fazla kip kullanarak iletişim kurulmasını sağlayan görsel dillerdir. İşaret dili tanıma modelleri, sağır ve duyma engeli bulunmayan insanlar arasında iletişimi kolaylaştırma potansiyeline sahiptir. Derin öğrenme alanındaki ilerlemeler ve erişime açık veri kümelerinin sayısının artması daha fazla araştırmacıyı işaret dili tanıma alanına yönlendirmiştir. Derin öğrenme alanındaki ilerlemeler ile işaret dili tanıma çalışmalarında kullanılan manuel öznitelik çözümlerinin yerini 2 boyutlu Evrişimsel Sinir Ağları (2B ESA) almaya başlamıştır. 2B ESA'nın zamansal modellemedeki yetersizliği ve 3B ESA'nın uzam-zamansal modelleme kabiliyeti 3B ESA'yı çok kullanılan bir çözüm haline getirmiştir. 3B ESA'ların başarılı sonuçlarına rağmen hesaplama maliyetinin ve hafıza ihtiyacının yüksek olması alternatif mimariler aranmasına sebep olmuştur. Bu tezde 2B ESA tabanlı zamansal kayma ile dikkat modellemesi yapan bir işaret dili tanıma modeli önerdik. 2B ESA kullanılması, karşılığı olan 3B ESA'ya göre parametre sayısını ve gerekli hafıza boyutunu azaltmıştır. Diğer veri kümeleri ile uygulanabilirliğini artırmak ve eğitim sürecini kolaylaştırmak için işaretçinin belirli vücut bölümlerine odaklanan görüntü kesimleri yerine tam çerçeve RGB görüntüler kullanılmıştır. İşaret dilinde iletişim birden çok görsel kipin aynı veya farklı zamanlarda kullanılması ile sağlandığı için model bu kiplerin birbirleri ile nasıl etkileşime girdiğini öğrenmelidir. Zamansal kayma modülleri 2B ESA tabanlı modele zamansal modelleme kabiliyeti verirken, dikkat modülleri ise videolarda neye, nereye ve ne zamana odaklanacağını öğrenmektedir. Modelimizi, Türkçe izole işaret dili veri kümesi olan BosphorusSign22k ile test ettik. Önerilen model %92.97 sınıflandırma başarımı elde etmiştir. Çalışmamız, izole işaret dili tanımada 2B ESA tabanlı zamansal kayma ile dikkat modellemesi yaparak rekabetçi sonuçlar alınabileceğini göstermiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS i
ABSTRACT ii
ÖZET iii
LIST OF FIGURES vi
LIST OF TABLES ix
LIST OF SYMBOLS xi
LIST OF ACRONYMS/ABBREVIATIONS xii
1. INTRODUCTION 1
2. RELATED WORK 5
2.1. Human Action Recognition 5
2.1.1. Human Action Recognition Datasets 9
2.2. Sign Language Recognition 12
2.2.1. Sign Language Recognition Datasets 16
3. ATTENTION MODELING WITH TEMPORAL SHIFT IN SIGN LANGUAGE RECOGNITION 20
3.1. Temporal Shift Module 20
3.2. Attention Modeling 24
3.2.1. Channel-Temporal Attention 26
3.2.2. Spatio-Temporal Attention 29
3.2.3. How Attention Works 31
4. EXPERIMENTS AND RESULTS 35
4.1. Dataset 35
4.2. Data Preprocessing and Transformations 36
4.3. Frame Selection 37
4.4. Temporal Shift Modules 38
4.5. Attention Modeling 42
4.5.1. Mature Feature Guided Regularization 43
4.6. Comparison With Other Studies 44

4.7. Attention Visualization 46
5. CONCLUSION 47
REFERENCES 49
APPENDIX A: USAGE OF FIGURES 59

LIST OF FIGURES

Figure 2.1. Example actions from Kinetics400 Dataset. 6

Figure 2.2. Example action from Something-Something Dataset. 7

Figure 2.3. Example actions from HMDB51 Dataset. From left to right action classes are: hand-waving and drinking. 9

Figure 2.4. Example actions from UCF101 Dataset. From left to right action classes are: rafting, cricket shot and shaving beard. 10

Figure 2.5. Example actions from Sports-1M Dataset. From left to right action classes are: track cycling and ultramarathon. 10

Figure 2.6. Example actions from YouTube-8M Dataset annotated with Guitar keyword. 11

Figure 2.7. Example actions from Kinetics400 Dataset from the "dunking basketball" class. 11

Figure 2.8. Example action from Something-Something Dataset that shows putting a white remote controller into a cardboard box. 12

Figure 2.9. Example signs from ASLLVD. From left to right signs are: wash and chew. 16

Figure 2.10. Example signs from CSL dataset. From left to right signs are: mother, father and thin. 17

Figure 2.11. Example signs from BosphorusSign22k. From left to right signs are: insurance and price. 17

Figure 2.12. Example from Signum dataset showing brother sign. 18

Figure 2.13. Example from RWTH-PHOENIX-Weather showing slippery sign. . 18

Figure 3.1. Visualization of model's input shape. 21

Figure 3.2. Illustration of how feature channels of consecutive video frames are shifted in Temporal Shift Module. 22

Figure 3.3. Visualization of the ResNet-18 model architecture. 23

Figure 3.4. Detailed visualization of the ResNet-18 model's building block with temporal shift operation. 24

Figure 3.5. Visualization of the proposed model's architecture. Overview of attention modules are shown at the top of the figure. 26

Figure 3.6. Detailed illustration of the channel-temporal attention submodule. 27

Figure 3.7. Detailed illustration of the spatio-temporal attention submodule. . 30

Figure 3.8. Illustration of teacher and student models' architecture in training process of student model. 34

Figure 4.1. Right most signer is used for testing, other ones are used for training. 35

Figure 4.2. Illustration of the image preprocessing and transformation pipeline used in the training process. 36

Figure 4.3. Illustration of different frame selection methods. From left to right: linear, segment and active frame selection. Green squares show selected 4 frames out of 16 in an example input video. 38

Figure 4.4. Visualization of the dropout layer inserted ResNet-18 + TSM architectures. From top to bottom: TSM with 3 dropout layers, TSM with 4 dropout layers. 41

Figure 4.5. Visualization of spatio-temporal attention regions. 46

LIST OF TABLES

Table 2.1. Summary table of human action recognition datasets. 12

Table 2.2. Summary table of isolated and continuous sign language recognition datasets. 19

Table 4.1. Comparison of ResNet-18 and ResNet-18 + TSM. 39

Table 4.2. Effects of using different horizontal flip probabilities tested with ResNet-18 + TSM. 39

Table 4.3. Effects of using different frame selection methods tested with ResNet-18 + TSM. 40

Table 4.4. Effects of selecting different number of frames from videos tested with ResNet-18 + TSM. 40

Table 4.5. Comparison of selecting different model architectures and batch sizes. 42

Table 4.6. Comparison of ResNet-18 + TSM with attention models. 43

Table 4.7. Effects of different α values in attention models with AGFR and MFGR. 43

Table 4.8. Effects of using different teacher model architectures in attention models with AGFR and MFGR. 44

Table 4.9. Comparison of our model with other studies used BosphorusSign22k dataset. 45

Table 4.10. Comparison of the proposed model with the state-of-the-art. 46

LIST OF SYMBOLS

$a^c()$	Channel-temporal attention function
$a^s()$	Spatio-temporal attention function
d_{c_avg}	Channel feature map generated by Average Pooling
d_{c_max}	Channel feature map generated by Max Pooling
d_{s_avg}	Spatial feature map generated by Average Pooling
d_{s_max}	Spatial feature map generated by Max Pooling
F	Output feature map of TSM blocks
F_{agm}	Refined feature map
F_c^i	Output feature map of ith channel-temporal attention module
F_s^i	Output feature map of ith spatio-temporal attention module
F_{agm}^s	Refined feature map of student model
F_{agm}^t	Refined feature map of teacher model
L_c	Cross-entropy loss
L_{mfgr}	Mature feature guided regularization loss
L_s	Student model loss
M	Mask
M_{c_frame}	Frame level channel attention mask
M_{c_video}	Video level channel attention mask
M_{meta}	Spatially aligned spatio-temporal attention masks
M_{frm}	Feature refining mask
M_{s_frame}	Frame level spatial attention mask
M_{s_video}	Video level spatial attention mask
y_i	Label of ith sample
$\hat{y}_i$	Predicted label for ith sample
α	Coefficient to blend loss function terms

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AAP	Adaptive Average Pooling
AGFR	Attention Guided Feature Refinement
ASL	American Sign Language
ASLLVD	American Sign Language Lexicon Video Dataset
BLSTM	Bidirectional Long Short-Term Memory
C3D	Convolutional 3D
CNN	Convolutional Neural Network
CSL	Chinese Sign Language
CTC	Connectionist Temporal Classification
DSL	German Sign Language
EM	Expectation Maximization
FMMNN	Fuzzy Min-Max Neural Network
HMDB51	Human Motion Database
HMM	Hidden Markov Model
HOF	Histogram of Oriented Flow
HOG	Histogram of Oriented Gradients
I3D	Two-Stream Inflated 3D CNN
IDT	Improved Dense Trajectory
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
KSL	Korean Sign Language
LSTM	Long Short-Term Memory
MBH	Motion Boundary Histogram
MC	Mixed Convolution
MEI	Motion Energy Image
MFGR	Mature Feature Guided Regularization
MHI	Motion History Image

MLP	Multi-Layer Perceptron
NGC	N-Gram Classifier
NLP	Natural Language Processing
ResNet	Residual Network
RNN	Recurrent Neural Network
SL	Sign Language
SLR	Sign Language Recognition
STMC	Spatial-Temporal Multi-Cue
SVM	Support Vector Machine
TAF	Temporal Accumulative Features
TMDHMM	Tied-Mixture Density Hidden Markov Model
TSM	Temporal Shift Module
TSN	Temporal Segment Network
WIC	Word-Independent Classifiers

1. INTRODUCTION

Sign languages are visual communication systems used by deaf communities as their primary language. Sign language users, signers, use multiple manual and non-manual visual cues to convey meaning in their communication with other signers. While the manual cues include shape, movement, position and orientation of hands, the non-manual cues include upper-body postures, head-shoulder movements and different kinds of facial expressions like eye gaze and mouth gestures [1].

Sign language recognition (SLR) refers to a field that aims at understanding and inferring signs performed by a signer in a recorded video. Essentially SLR is a video classification task; however, it has challenging characteristics. Firstly, SLR models recognize non-manual gestures together with manual ones in order to understand the performed sign. Therefore, the model must be able to capture fine differences in human pose. Secondly, SLR models use information gathered from human pose and movement while other video classification problems can also benefit from context information in videos. Another challenge is that since standardization of sign languages is not very common, different signers can perform different gestures to execute the same sign. Moreover, at times the same signer can skip some of the visual cues for a given gloss depending on performed signs previously [2]. The ground truth for individual signs is assigned by experts, by assigning labels called "glosses", which are words from the spoken language.

Sign language recognition tasks can be divided into two categories according to the problem they aim to solve. The first category, isolated SLR problem, tries to recognize words or phrases corresponding to a specific sign performed in video. All visual cues to express signs are presented in the video and each video contains only one sign. Proposed solutions for this category must be able to distinguish differences between many signs. The second category, continuous SLR problem, aims to identify and recognize signs that appear in sequence. The sequence of signs appears in videos

in a continuous manner. Therefore continuous SLR models must be able to recognize the start and end point of each sign performed by the signer in addition to challenges in isolated SLR.

Sign languages are not universal and they are also different from corresponding spoken languages in terms of grammar and vocabulary. These differences make communication between deaf communities and hearing people even harder. Automated systems that understand and translate a sign language to a spoken language can alleviate communication problems between two communities. SLR has been an active research domain for over 30 years [3]. We conducted our study with a Turkish isolated sign language dataset in order to present an effective and efficient solution that can be helpful for other researchers and deaf community in Turkey. Moreover, the presented architecture can easily be applied for other sign languages in the world.

Early studies in SLR used wearable sensors to collect data and extract features. Kadous [4] used instrumented gloves to extract features from signers' hands. Covered distance and time duration to perform a sign together with histograms for wrist rotations, finger bends and positions in 3D coordinates are used as features in order to recognize the corresponding sign.

Hand-crafted feature based methods were also used in the SLR domain. Liwicki and Everingham [5] used videos taken by consumer quality webcam. They used the Histogram of Oriented Gradients (HOG) descriptor to represent hand shape. For temporal modeling of hand-crafted features Hidden Markov Models (HMM) are used in different studies [5, 6].

Thanks to advancements in deep learning and new public datasets, neural networks achieved high performances in both image [7, 8] and video [9, 10] tasks in the computer vision domain. Studies in SLR have also started to use deep learning models like other tasks in computer vision [1, 3].

Deep learning based models have been used in isolated SLR. Özdemir *et al.* [11] used 3D residual network (3D ResNet) architecture with mixed 2D-3D convolution layers proposed in [12]. Gökçe *et al.* [13] also used 3D ResNet with mixed 2D-3D convolution layers. They trained separate models for different visual cues and fused their outputs to classify signs. Vázquez-Enríquez *et al.* [14] used multiscale spatial-temporal graph convolutional network for isolated SLR.

The resemblance of continuous SLR to the daily communication problem between deaf and hearing people in the real world attracts researchers. Koller *et al.* [15] proposed an architecture of 2D CNN with HMM for continuous SLR. Camgoz *et al.* [16] proposed a novel architecture with CNN and Bidirectional Long Short-Term Memory (BLSTM) layers for spatial and temporal modeling while using Connectionist Temporal Classification (CTC) loss. Zhou *et al.* [1] proposed a spatial-temporal multi-cue (STMC) that uses 2D CNN to extract spatial features from different cues and deconvolutional layers for pose estimation. It models temporal information by using BLSTM and CTC.

This thesis aims to achieve competitive performance in isolated SLR without suffering from high computational costs and memory requirements. For this purpose, we presented a network with a 2D CNN backbone that lowers memory and computational costs. Temporal shift modules (TSM) are inserted into the backbone in order to compensate for insufficiency of 2D CNN models in temporal modeling. Since we only use one model to predict the class of a sign, the model must be able to focus on important regions and moments of performed visual cues. Attention modules with channel-temporal attention and spatio-temporal attention submodules perform this task. Channel-temporal attention focuses on what is important in a frame and when that body part of movement is important in the sign video. On the other hand, spatio-temporal attention focuses on which area is discriminative in a frame and when that position is important in video.

The following chapters of the thesis are organized as follows. In Chapter 2, related works in sign language recognition and published SLR datasets are introduced. Chapter 3 provides detailed information about our model architecture and how each subunit in the model works. In Chapter 4, experimental results are presented and discussed. Chapter 5 contains conclusions and suggestions for future work.

2. RELATED WORK

Understanding video content has gained more importance with the increase in the number of videos. Video classification is a domain working for this purpose. It aims to label videos according to their contents. Human action recognition and sign language recognition are subdomains that are named according to the contents of videos they examine. While human action recognition focuses on human interactions with objects and other people, sign language recognition focuses on performed visual cues to express a sign gloss. Human action recognition uses more general data from a wider domain with coarse motions relative to sign language recognition. Therefore datasets and proposed solutions in the field of human action recognition can be adapted to sign language recognition.

2.1. Human Action Recognition

Human action recognition is the task of recognizing human actions in videos. These actions generally involve motion of a single person, interaction of multiple people or interaction of humans with objects (see Figures 2.1 and 2.2 for examples). It is an interesting subject because proposed solutions can be used in other real world applications including human-computer interaction, video retrieval, gaming and entertainment.

Even though recognizing an action is simple for a human, it is challenging for computer systems because of several reasons. Firstly, cluttered backgrounds in the videos may contain distracting objects or irrelevant human actions [17]. Secondly, human action videos have both interclass and intraclass variations [18]. Another challenge is that developing complex models capable of recognizing many actions has high computational cost.

Early studies in the field used hand-crafted features for representation and machine learning models for classification. Bobick and Davis [19] presented Motion History Images (MHI) to represent the accumulated trajectory of moving objects. It is an enhanced version of Motion Energy Images (MEI) [20] which is used to show regions of motion. Laptev *et al.* [21] used Histogram of Oriented Gradient (HOG) and Histogram of Oriented Flow (HOF) to represent motion. They used a nonlinear Support Vector Machine (SVM) for classification.



Figure 2.1. Example actions from Kinetics400 Dataset.

Wang *et al.* [22] used HOG, HOF and Motion Boundary Histogram (MBH) to extract features from dense trajectories. After the success of dense trajectories in action recognition, improved dense trajectory (IDT) is introduced in [23]. It aims to improve trajectory features by deleting camera motion trajectories and canceling out camera motion from optical flow. IDT used Fisher Vectors [24] for feature encoding and SVM for classification.



Figure 2.2. Example action from Something-Something Dataset.

Krizhevsky *et al.* [25] presented a 2D CNN image classification model for ImageNet Large-Scale Visual Recognition Challenge 2012 (ILSVRC) [26]. The success of this model attracted researchers to design deep learning models for both image and video related computer vision problems. Faster and specialized hardwares, publication of available large datasets and advancements in deep learning have shifted the main focus in human action recognition to deep learning models.

Karpathy *et al.* [9] presented an early study which applies 2D CNN model to the action recognition domain. The study investigated different models to fuse temporal information and to speed up the training process. Firstly, it proposed a 2D CNN model that takes a single frame at a time to predict the action in the video. Secondly, it examined several methods to fuse temporal information. Combining frames from a time window at pixel level or training two seperate networks with joint classification layer with multiple frames were tried options. Moreover, it proposed a two stream model architecture that processes low and high resolution images of the same frame in its streams. This design lowers training time of the network without decreasing its accuracy.

Simonyan and Zisserman [10] proposed a two-stream CNN architecture for action recognition in videos. The first one, spatial stream, takes static images from input video. This stream is especially useful for predicting actions related to an object. The second one, temporal stream, uses dense optical flow which represents motion between consecutive frames as input. Softmax outputs of two streams are fused by averaging

or SVM. Separating the network to two streams gives the chance to use pre-trained networks with large datasets in the spatial stream. The architecture is related to the two-stream hypothesis in the human visual system [10]. The spatial stream is similar to the ventral stream in humans which recognizes objects and the temporal stream is related to the dorsal stream which recognizes motion. Ng *et al.* [27] also used a two stream CNN architecture with raw frames and optical flow inputs to compute feature maps. These frame-level feature maps are aggregated by Long Short-Term Memory (LSTM) and video level action predictions are made. Wang *et al.* [28] proposed Temporal Segment Networks (TSN), a two-stream architecture that can learn long-range temporal information. The idea behind the solution is that consecutive frames have very similar content so picking them to represent action is unnecessary. Instead of this strategy, [28] suggested to divide the video into segments. One frame from each segment is selected and given into the model as input. Frame level predictions from the spatial stream and the temporal stream are aggregated separately by corresponding segmental consensus functions. Results of two segmental consensus are fused and video level prediction is obtained.

Tran *et al.* [29] introduced a 3D CNN model named Convolutional 3D (C3D) to learn spatio-temporal features and classify action recognition videos. The downside of the network is having more parameters than a 2D CNN, making the network harder to train. Carreira and Zisserman [30] proposed converting successful 2D CNN models to 3D CNN models. The first step is inflating all filters and pooling kernels to add a temporal dimension. Then, in order to benefit from huge image datasets like ImageNet, weights of the selected pre-trained 2D CNN can be bootstrapped to a 3D CNN model. They introduced a two-stream inflated 3D CNN (I3D) that uses both RGB and optical flow as inputs.

Both 3D CNN and two-stream models have efficiency problems when trained with big datasets. 3D CNN models have a high number of parameters which makes them hard to train and two-stream models require computing and storing optical flow information before the training process. These reasons cause search for alternative

efficient methods. Lin *et al.* [31] introduced Temporal Shift Module (TSM) that shifts part of the feature channels obtained by different frames. This operation enables a 2D CNN to model temporal information without adding more parameters to the backbone model.

2.1.1. Human Action Recognition Datasets

Deep learning models usually give better results when trained with large datasets. This created a need for larger action recognition datasets to improve model performances, to provide challenging datasets for more complex models and to create new benchmarks. Summary information about human action recognition datasets can be found in Table 2.1.

Kuehne *et al.* [32] introduced the Human Motion Database (HMDB51) in 2011. The dataset has approximately 7000 videos from 51 action categories and each category has at least 101 clips. The videos are collected from YouTube, Google, digitized movies and public databases (see Figure 2.3).



Figure 2.3. Example actions from HMDB51 Dataset. From left to right action classes are: hand-waving and drinking.

Soomro *et al.* [33] introduced UCF101, the largest human action dataset at its time, in 2012. It has 13320 videos from 101 action groups. The videos are collected from YouTube (see Figure 2.4).



Figure 2.4. Example actions from UCF101 Dataset. From left to right action classes are: rafting, cricket shot and shaving beard.

Karpathy *et al.* [9] introduced the Sports-1M dataset in 2014. The dataset has one million YouTube videos from 487 classes. Each class has between 1000 to 3000 related videos (see Figure 2.5).



Figure 2.5. Example actions from Sports-1M Dataset. From left to right action classes are: track cycling and ultramarathon.

YouTube-8M dataset [34] was published in 2016. It has around 8 million videos with a duration of 500K hours in total. The videos have multi-labels and these labels are taken from YouTube video annotation system (see Figure 2.6).



Figure 2.6. Example actions from YouTube-8M Dataset annotated with Guitar keyword.

Kinetics400 dataset [35] was published in 2017. The dataset has videos from 400 different action categories and each of these categories has between 400 and 1150 clips. The dataset has over 300K videos collected from YouTube platform. It contains videos of single person actions, person-person actions and person-object actions (see Figure 2.7).

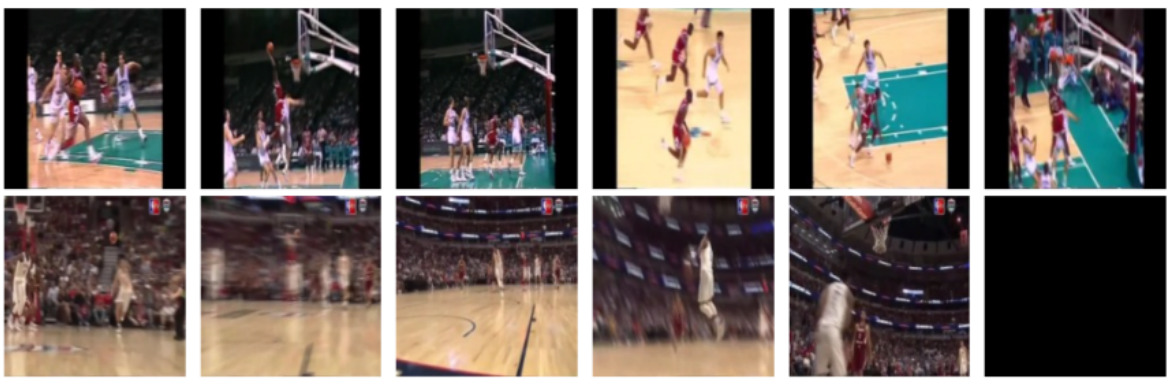

Figure 2.7. Example actions from Kinetics400 Dataset from the "dunking basketball" class.

The first version of the something-something dataset [36] was introduced in 2017. It is a fine-grained action recognition dataset and has 108499 videos from 174 action classes. The videos contain human-object interactions and are labeled with textual descriptions (see Figure 2.8).



Figure 2.8. Example action from Something-Something Dataset that shows putting a white remote controller into a cardboard box.

Table 2.1. Summary table of human action recognition datasets.

Dataset	Year	# Classes	# Videos
HMDB51 [32]	2011	51	$\sim$ 7000
UCF101 [33]	2012	101	13,320
Sports-1M [9]	2014	487	1 million
YouTube-8M [34]	2016	4800	8,264,650
Kinetics400 [35]	2017	400	306,245
Something-Something [36]	2017	174	108,499

2.2. Sign Language Recognition

Sign language recognition (SLR) refers to the task of inferring the label of a performed sign by examining manual and non-manual visual cues in a video. These cues mainly consist of hand gestures, upper-body movements and facial expressions of the signer. Human experts can comprehend collaboration between visual cues. These visual cues can happen at the same time or in sequence. Moreover some of them are fine-grained movements and temporal relationships between them have critical

importance to recognize the corresponding gloss. SLR models should have the ability to understand spatial and temporal relations between visual cues.

SLR models are proposed to work with isolated or continuous SL datasets. Isolated SL datasets consist of videos that contain visual cues to express a single gloss. The gloss corresponds to a word or a phrase in spoken language. On the other hand, continuous SL datasets have videos showing a sequence of glosses in a continuous manner. Moreover, SLR has subdomains that are related to hand signs and motions. These are hand detection, hand pose estimation, hand gesture recognition, real-time hand tracking and hand pose recovery [37].

Early studies in the SLR domain make use of wearable sensors. These sensors are used to collect informative data regarding a signer's hand positions and movements. Vogler and Metaxas [38] used the Ascension Flock of Birds which is an equipment with sensors and a magnet. Three dimensional wrist positions and orientations are collected with this equipment. The collected data is modeled with an HMM and presented as a solution for continuous American Sign Language (ASL) recognition problem. Kadous [4] used PowerGlove, an instrumented glove, to get position data, wrist rotations and finger bends. The data is used for feature extraction. Lee *et al.* [39] proposed an approach for Korean Sign Language (KSL) recognition. They used CyberGlove and Polhemus sensor to acquire finger flexures, hand positions and orientations. The proposed approach used fuzzy rules for direction classification and Fuzzy Min-Max Neural Network (FMMNN) to classify the posture and orientation of a signer.

Wearable sensors were used and maintained their importance in the SLR domain until 2005 [40]. Their dominance in SLR was replaced by vision based data such as RGB and depth data after 2005. In the transition period from wearable sensors to vision based data, colored gloves were also used. Colored gloves, as the name suggests, are colorful gloves that help researchers to detect signer's hand in video frames. Zhang *et al.* [41] introduced an architecture to recognize Chinese Sign Language (CSL). They used colored gloves to detect the singer's hands in a video taken by a USB camera.

Detected hands are used to extract features from the dominant hand, non-dominant hand and finger area of the dominant hand. Tied-Mixture Density Hidden Markov Model (TMDHMM) is used for sign recognition.

RGB and depth data are two main input types in vision based data used for SLR. RGB data has become the most used data type in SLR since 2005. Also the number of studies with depth data increased after 2010 thanks to the Kinect sensor. RGB videos generally have high resolution image content and depth data contains information about distance in videos [37].

Usage of visual data prompts researchers to extract features from RGB videos or depth data. Early studies with visual data extract hand-crafted features then generally use two models: one to understand the temporal relation in video and another one to classify the performed sign gloss.

Wong and Cipolla [42] focused on classifying 10 primitive hand motions. Video frames are processed with MHI and MEI operations. Their outputs are used to compute Motion Gradient Orientation (MGO) images. Motion features are extracted from MGO images and classified by a sparse Bayesian classifier. In ChaLearn Looking at People Challenge 2014 [43], a winning solution for gesture recognition used hand-crafted features. The solution extracted HOG features, skeletal joint position and distance features. Camgöz *et al.* [44] proposed SLR system to help deaf people come to a hospital. The proposed system uses a Kinect sensor to collect signer's data. Features extracted for hand position, hand movement, hand shape, upper body pose and Principal Component Analysis (PCA) are applied for each feature to combine them. Dynamic Time Warping (DTW) and Temporal Templates (TT) are used to model temporal information. K-Nearest Neighbors (KNN) and Random Decision Forest (RDF) are used for classification. Özdemir *et al.* [45] introduced an architecture for isolated SLR. Firstly, IDT [23] features are extracted. Then, the proposed method uses Gaussian Mixture Model (GMM) and PCA to increase efficiency and decrease sizes of feature vectors. These features are represented by Fisher Vectors [24] and SVM is used to classify them.

Advancements in deep learning model architectures and techniques, improvements of specialized hardwares especially in GPUs, publication of large and high-quality available datasets attracts researchers to propose deep learning based solutions to video related tasks. Deep learning models are proposed for human action recognition [10, 29] and gesture recognition [46]. Researchers also introduced deep learning solutions for SLR due to its similarity with other domains.

Neverova *et al.* [47] worked on hand gesture recognition from depth images to improve human-computer interactions. The study proposed a 2D CNN model trained on depth labeled synthetic hand gestures images and unlabeled real depth images taken from consumer level depth sensors. Koller *et al.* [48] proposed a model architecture for continuous SLR by combining a CNN with HMM and achieved state of the art results on three datasets, RWTH-PHOENIX-Weather 2012 [49], SIGNUM single signer [50] and RWTH-PHOENIX-Weather 2014 Multisigner [51]. Koller *et al.* [52] proposed using parallelism for continuous SLR. The introduced architecture trains multiple CNN-LSTM models with full frame inputs and based on their loss functions they become sign language, hand shape and mouth shape classifiers. Outputs of these classifiers are improved by an HMM in several Expectation Maximization (EM) iterations.

3D CNN based architectures were adopted in the SLR domain due to their spatio-temporal modeling abilities. Joze and Koller [53] proposed using I3D [30], a model introduced for action recognition, to recognize isolated ASL. Wei *et al.* [54] introduced a new architecture for continuous SLR. 3D ResNet, BLSTM and global temporal pooling are used at early stages of the architecture. While 3D ResNet extracts spatio-temporal features, BLSTM learns contextual relationships in videos. Global temporal pooling takes outputs of BLSTM and produces fixed length feature vectors to represent video content. Word-independent classifiers (WIC) take the feature vectors and recognize words. The architecture also has an n-gram classifier (NGC) for sentence recognition. Gökçe *et al.* [13] proposed using multiple 3D ResNet models with mixed 2D-3D convolution layers for Turkish isolated SLR. Each of these cue models are specialized in different regions, namely, hand, upper-body and face. Each model makes its predictions

independently. A weighted fusion algorithm based on models' success on validation sets is used to fuse output probabilities of cue models and make final class predictions.

2.2.1. Sign Language Recognition Datasets

Sign languages differ from each other in different countries like the spoken language. Therefore an SLR model for a specific language needs to be trained with a SL dataset of that language. Large scale SL datasets are needed to increase model success and to teach complex structures to the deep learning models [37]. Another important characteristic of SL datasets besides language is the annotation type of the dataset. Isolated SL datasets have gloss level labels and continuous SL datasets have sentence level labels [2]. Summary information about SLR datasets can be found in Table 2.2.

Neidle *et al.* [55] introduced American Sign Language Lexicon Video Dataset (ASLLVD) in 2012. The dataset is developed by computer scientists and linguists. It is an isolated ASL dataset with 2742 sign classes and 9794 video clips performed by 6 signers. 4 synchronized video cameras were used to collect data (see Figure 2.9).



Figure 2.9. Example signs from ASLLVD. From left to right signs are: wash and chew.

Zhang *et al.* [56] introduced an isolated CSL dataset with 500 sign classes and 2500 video clips performed by one signer. The dataset is collected with Kinect v2. It is used to get RGB data and 3-D coordinates of skeleton points (see Figure 2.10).



Figure 2.10. Example signs from CSL dataset. From left to right signs are: mother, father and thin.

Joze and Koller [53] published isolated MS-ASL datasets in 2019 with four subsets. The biggest subset contains 25513 videos from 1000 sign classes performed by 222 signers. The dataset is signer independent and collected in unconstrained conditions.

Özdemir *et al.* [11] published BosphorusSign22k which is a Turkish isolated SL dataset in 2020. The dataset has 744 sign classes and 22542 videos performed by six signers. The dataset is recorded with Kinect v2 and contains RGB videos, depth map and skeleton information (see Figure 2.11).

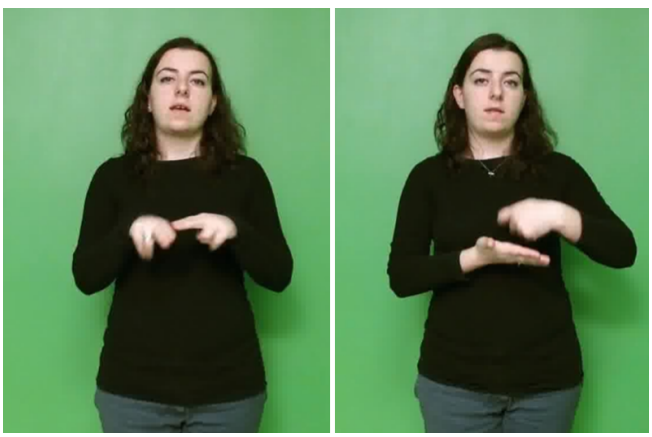

Figure 2.11. Example signs from BosphorusSign22k. From left to right signs are: insurance and price.

Agris *et al.* [50] published the SIGNUM dataset in 2008. The dataset has both isolated and continuous sign videos for German Sign Language (DSL). 25 singers performed 455 basic signs and about 19k sentences (see Figure 2.12).



Figure 2.12. Example from Signum dataset showing brother sign.

Forster *et al.* [49] introduced the RWTH-PHOENIX-Weather dataset. It is collected from a weather forecast programme of Phoenix which is a German TV station. It is a continuous DSL dataset performed by seven signers. The dataset has over three hours of data showing 1980 sign sentences and 911 distinct glosses (see Figure 2.13).



Figure 2.13. Example from RWTH-PHOENIX-Weather showing slippery sign.

Huang *et al.* [57] introduced a modern CSL dataset for continuous SLR. It consists of 25k videos annotated with a complete sentence and total video duration is over 100 hours. Videos in the dataset are performed by 50 signers and recorded with Kinect. Data for depth and body joints are recorded in addition to RGB videos.

Table 2.2. Summary table of isolated and continuous sign language recognition datasets.

Dataset	Year	Language	Type	# Signers	# Classes - Vocabulary	# Videos
ASLLVD [55]	2012	English	Isolated	6	2742	9794
CSL [56]	2016	Chinese	Isolated	1	500	2500
MS-ASL-1000 [53]	2019	English	Isolated	222	1000	25,513
BosphorusSign22k [11]	2020	Turkish	Isolated	6	744	22,542
SIGNUM [50]	2008	German	Continuous	25	455	$\sim 19,000$
RWTH-PHOENIX-Weather [49]	2012	German	Continuous	7	911	-
CSL [57]	2018	Chinese	Continuous	50	-	$\sim 25,000$

3. ATTENTION MODELING WITH TEMPORAL SHIFT IN SIGN LANGUAGE RECOGNITION

We have applied attention modeling with temporal shift [31] in order to recognize isolated sign language videos. Attention modeling modules and temporal shift modules (TSM) are added on top of the 2D CNN backbone. TSM enables 2D CNN to model temporal information. Attention modeling enables the model to focus on important and distinctive body parts, motions, places and times in the video. We chose the ResNet-18 [7] model as the backbone but other popular 2D CNN models can also be used as alternatives.

This 2D CNN based model architecture used for video classification takes inputs with the same shape used in image classification models. Input shape is $T \times C \times H \times W$ where T is batch size times number of frames per video, C is the number of channels (3 for RGB images), $H \times W$ is the spatial size of frames (see an example in Figure 3.1).

3.1. Temporal Shift Module

Advancements in deep learning techniques, improvements of specialized hardwares and publication of large datasets have made deep learning models the standard solution for video understanding tasks such as action recognition and sign language recognition [17,31] . Early studies with deep learning in this field focused on 2D CNN models [9, 10]. Classic 2D CNN models learn spatial features but they can not learn temporal information well enough. This issue decreases models' success on video tasks. It causes studies to use 3D CNN networks because they can learn spatio-temporal features [29, 30, 58]. Thanks to the ability to capture motion information from adjacent frames, 3D CNN networks are widely used [58]. Despite its capabilities, 3D CNN networks require high available memory, also have high computation cost and lots of parameters. These downsides forced researchers to find alternative solutions [31, 59].

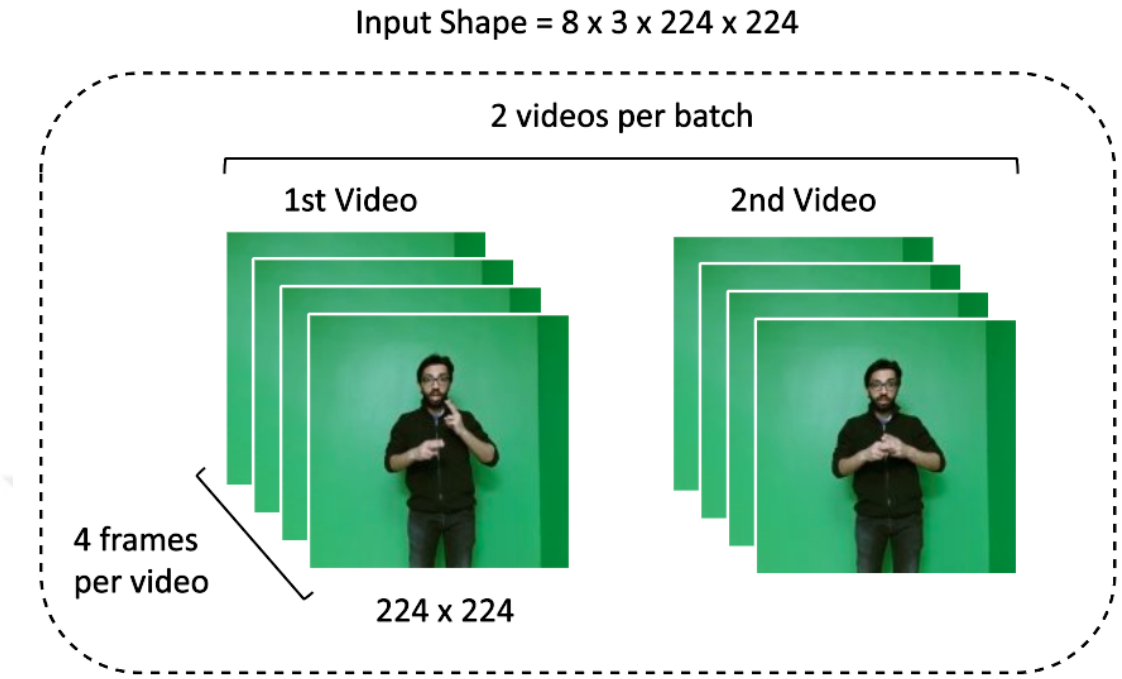

Figure 3.1. Visualization of model's input shape.

TSM was introduced as one of these solutions and managed to get successful results in the action recognition domain. It uses 2D CNN models as backbone instead of 3D CNNs because of their lower computational cost and memory need. 2D CNN models can not model temporal information in videos without using extra preprocessed inputs that represent motion between consecutive frames (e.g. optical flow) or trainable submodels developed to be used with sequential data (e.g. Recurrent Neural Networks (RNN), LSTM). The strength of TSM is that it provides the ability of temporal modeling without adding extra parameters to the backbone networks. Normally, 2D CNN networks used in the video domain take frames one by one, extract feature channels from each frame independently and give an output accordingly. On the contrary, when a 2D model with TSM takes one frame and extracts feature channels it shifts some of these channels with ones extracted from previous and following frames (see Figure 3.2). A 2D model with TSM is informed about motions in different moments when it gives an output for a frame. In this way, it can model temporal information in videos.

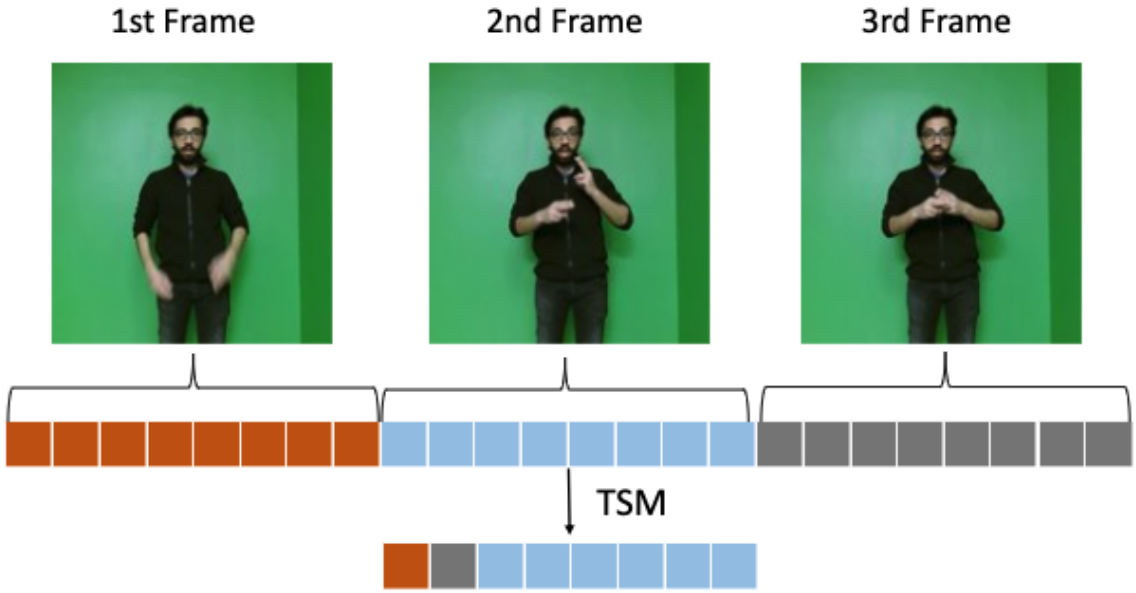

Figure 3.2. Illustration of how feature channels of consecutive video frames are shifted in Temporal Shift Module.

Lin *et al.* [31] used the ResNet-50 [7] model as the backbone network. On the other hand, we used the ResNet-18 model (see Figure 3.3 for architecture) as the backbone in order to reduce memory need, number of parameters and training time. The ResNet-18 model has 4 stages and 8 building blocks. A shift module is added before the first CNN layer of each of these building blocks (see Figure 3.4). We placed the shift module in the residual branch of the building block as advised in [31]. In this way, the model can still use original feature channels because they are transferred as unchanged via identity mapping.

Feature channels of different frames in video produced by previous parts of the network are given as input to the shift operation in TSM. First one eighth of the feature channels of a frame is replaced with corresponding channels of the previous frame. Second one eighth of the feature channels of a frame is replaced with corresponding channels of the following frame. In this way, the 2D CNN model gets temporal modeling ability. Thanks to the shift operation, the model gets temporal modeling ability without introducing new trainable parameters.

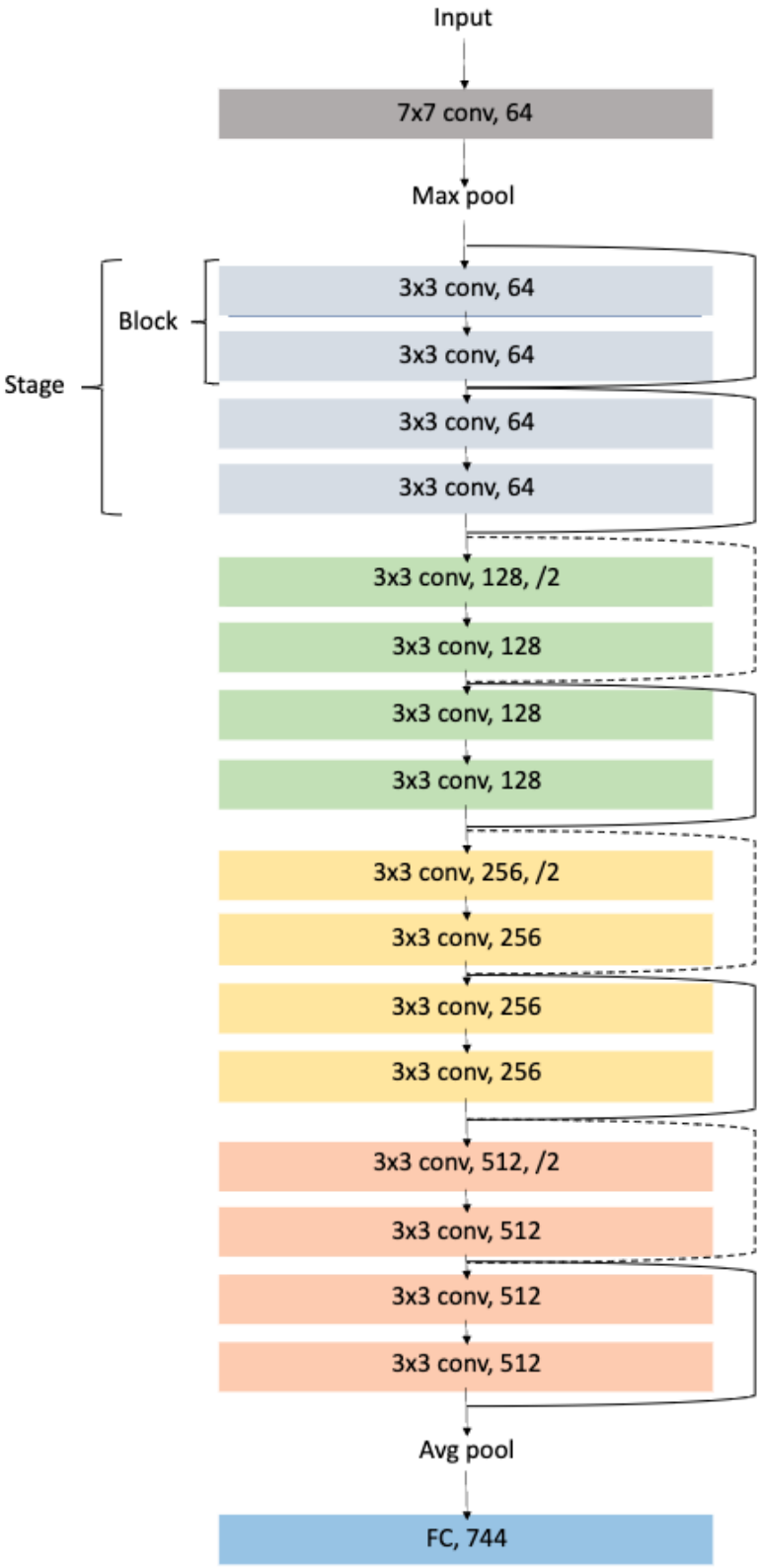

Figure 3.3. Visualization of the ResNet-18 model architecture.

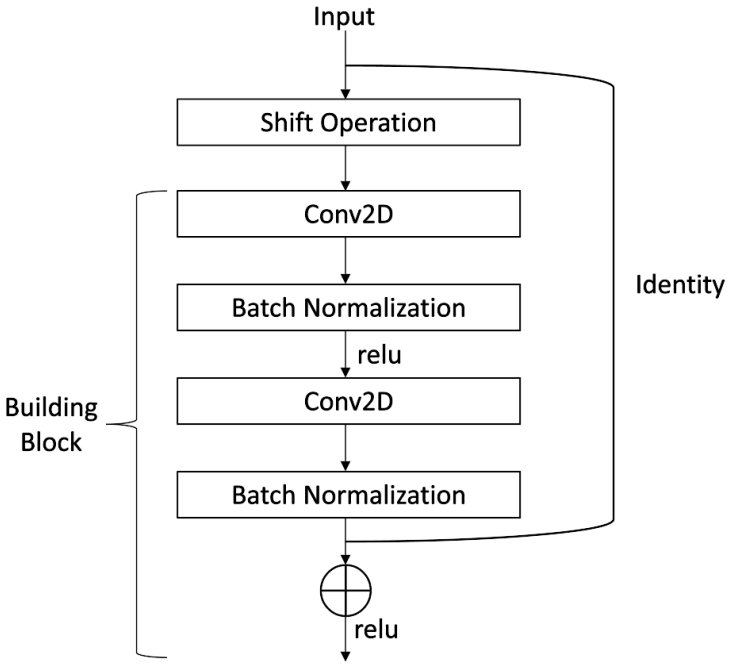

Figure 3.4. Detailed visualization of the ResNet-18 model's building block with temporal shift operation.

3.2. Attention Modeling

CNN models learn filters to detect both low-level and high-level features in an input image. These filters are traversed and applied to every part of an input image. When a filter that detects a specific feature finishes traversing the input image, the feature located anywhere in the image will be extracted. Model uses information gathered from feature occurrences all over the input image to interpret the scene in the visual data.

Characteristics of convolution operation may cause it to give redundant or unnecessary information to the model because not all occurrences of a feature are helpful to understand the input. This problem has encouraged researchers to develop new model architectures that can produce better representations [60].

Human visual system has an attention mechanism to solve this specific problem. Humans focus on the most important objects, motions or moments in the scene to understand what they see. This mechanism also inspired researchers to adopt attention into computer vision problems. It enables models to focus on important, distinctive parts of features and suppress unnecessary parts [61]. Attention mechanism combined with deep learning models are used in image [61,62] and video [17,63] related computer vision tasks.

Attention modules must learn to focus on different aspects of information according to the input data type. In natural language processing (NLP) studies [64], attention modules focus on what and when dimensions of the data. In computer vision tasks with 2D images, attention is on what and where dimensions. Attention modules have two dimensions to focus in both of these study areas. On the other hand, when working with video data attention modules must focus on three dimensions: what, where and when.

The addition of a third dimension into attention modeling causes challenges in model design and training process [17]. Wang *et al.* [65] showed that self-attention related non-local models achieved competitive results in video classification. However, non-local models come with a considerable computational cost [17]. Dhingra and Kunz [63] introduced a 3D residual attention network for hand gesture recognition in videos. The model uses 3D ResNet as its backbone and adds three attention blocks with different complexities. These attention blocks also include 3D CNNs. Using an architecture with many 3D CNN layers causes this model to have an extremely high number of parameters. A new attention module that can be used with 2D CNNs is introduced by Pérez-Rúa *et al.* [17]. The proposed module enables the model to perform attention in all three dimensions of the input data without introducing significant computational cost.

Attention modules take feature maps (F) as inputs. Shape of feature maps, like input video frames, is $T \times C \times H \times W$ (see an example in Figure 3.1). When

considered from a wider perspective, attention modules aim to generate masks (M) that have the same size with feature maps. Masks are element-wise multiplied with the corresponding feature maps. With this multiplication operation, effects of features that are important and distinctive are emphasized while irrelevant ones are suppressed [17]. These attention modules are placed after each stage of the ResNet-18 model. Each module has two submodules respectively: channel-temporal attention and spatio-temporal attention (see Figure 3.5 for model architecture).

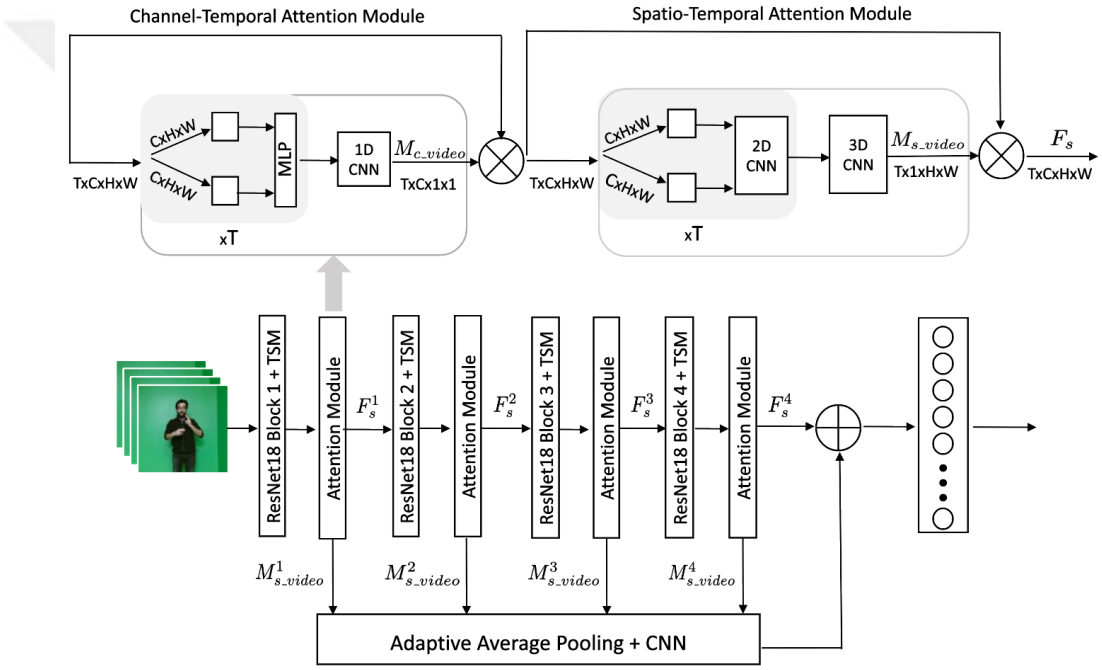

Figure 3.5. Visualization of the proposed model's architecture. Overview of attention modules are shown at the top of the figure.

3.2.1. Channel-Temporal Attention

The first submodule of attention, channel-temporal attention, enables the model to focus on information in what and when dimensions of input videos respectively. The submodule finds the importance of objects, motions and how that importance is changing over time. Architecture of the channel-temporal attention submodule can be seen in Figure 3.6.

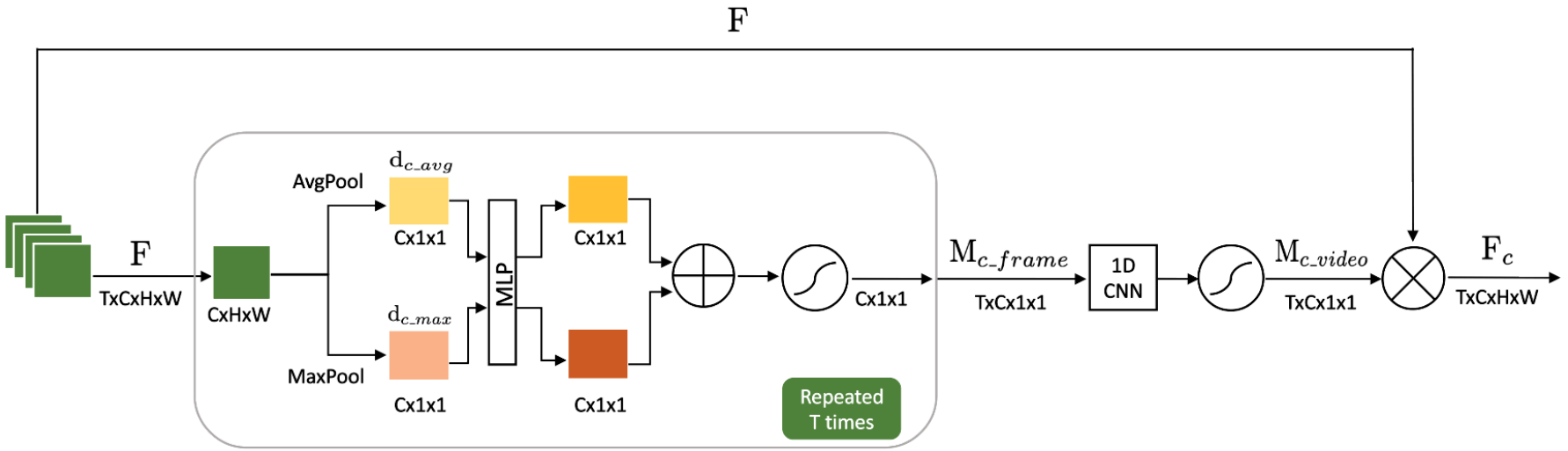

Figure 3.6. Detailed illustration of the channel-temporal attention submodule.

The first step in the submodule is squeezing the spatial dimension of feature maps in order to learn what to attend to in an efficient way. Average pooling is used commonly to aggregate spatial information in feature maps. Hu *et al.* [60] showed that average pooling is more successful for squeezing spatial dimensions compared to max pooling. Furthermore, Zhou *et al.* [66] used average pooling to get the extent of an object. However, Woo *et al.* [61] proposed using max pooling in addition to average pooling. They suggested that using max pooling will be beneficial because it can extract most distinctive features in the spatial dimension [61]. In our study, feature maps for each video frame with size $C \times H \times W$ are squeezed with average pooling and max pooling operations.

These operations generate two channel descriptors (d_{c_avg} and d_{c_max}) with size $C \times 1 \times 1$. The channel descriptors are given as input to a multi-layer perceptron (MLP) network with one hidden layer. Outputs of MLP for both channel descriptors are element-wise summed and processed with a sigmoid function to generate frame-level channel attention mask (M_{c_frame}) with size $C \times 1 \times 1$.

These steps are repeated for feature maps of each frame and their outputs are concatenated. The concatenation forms a tensor with size $T \times C \times 1 \times 1$ and it contains frame-level channel attention masks. These masks have coefficients that emphasize or suppress channels of feature maps of corresponding frames in the video. In other words, frame-level channel attention masks give models the ability to learn what to attend.

Frame-level channel attention masks are not sufficient for video tasks because they do not take temporal relationships into account. A CNN model with two layers of 1D convolution takes a tensor of frame-level channel attention masks as input. This model discovers temporal relations in corresponding channels between different frames. Output of the CNN model is further processed with a sigmoid function. The final output is video-level channel attention mask (M_{c_video}). Video-level channel attention mask contains coefficients to modify feature maps of the corresponding video. The mask has size $T \times C \times 1 \times 1$ and enables the model to learn what and when to attend.

3.2.2. Spatio-Temporal Attention

Second submodule of attention, spatio temporal attention, enables the model to focus on information in where and when dimensions of videos, respectively. The submodule finds important and informative regions and how they are evolving over time. The architecture of the submodule can be seen in Figure 3.7.

The first step in the submodule is squeezing the channel dimension of feature maps in order to learn where to attend. Average pooling and max pooling operations are applied to squeeze channel dimension of feature maps with size $C \times H \times W$. Similar to what happens in the channel-temporal attention module, these pooling operations generate two feature descriptors (d_{s_avg} and d_{s_max}). The difference is that since pooling is applied to squeeze the channel dimension, resulting feature descriptors have size $1 \times H \times W$. The two descriptors are concatenated and given to a model with one layer of 2D convolution. Output of this model is further processed with a sigmoid function. The final output is frame-level spatial attention mask (M_{s_frame}) with size $1 \times H \times W$.

Steps to generate frame-level spatial attention masks are repeated for feature maps of each frame in video. Resulting masks are concatenated to form a tensor with size $T \times 1 \times H \times W$. Each of these masks contain coefficients that emphasize or suppress spatial information in feature maps of corresponding frames in the video. In this way, frame-level spatial attention masks give models the ability to learn where to attend.

These frame-level spatial attention masks are computed without considering temporal information. Therefore, they are not suitable to be used with video data. Another model with two layers of 3D convolution is used to learn temporal relations between frame-level spatial attention masks of different frames. Another sigmoid function is applied to the output of the model. Resulting tensor with size $T \times 1 \times H \times W$ is a video-level spatial attention mask (M_{s_video}). The tensor has coefficients to modify spatio-temporal dimensions of feature maps of the video and enables the model to learn where and when to attend.

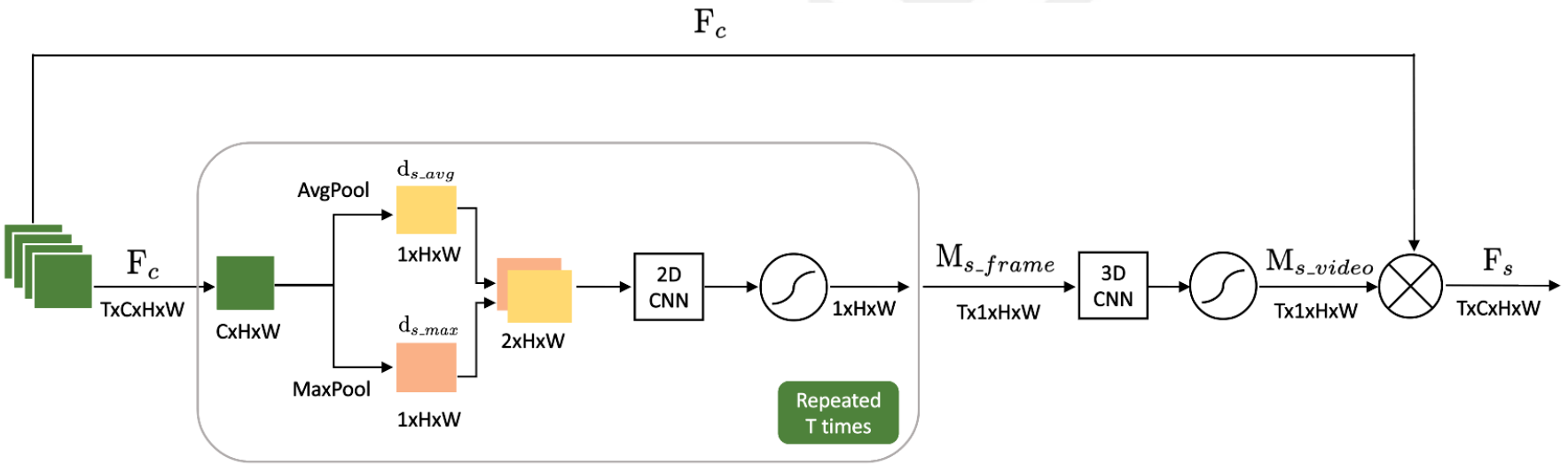

Figure 3.7. Detailed illustration of the spatio-temporal attention submodule.

3.2.3. How Attention Works

Four attention modules are placed after each residual building block with temporal shift. These building blocks output feature maps (F) with size $T \times C \times H \times W$. Attention modules apply channel-temporal attention and spatio-temporal attention, respectively. Thanks to channel-temporal attention module, channel attention mask (M_{c_video}) with size $T \times C \times 1 \times 1$ is produced. Since corresponding feature map (F) has spatial dimensions $H \times W$, channel attention mask is broadcasted during element-wise multiplication operation and outputs the modified feature map (F_c).

Spatio-temporal attention takes F_c as input and outputs spatial attention mask (M_{s_video}) with size $T \times 1 \times H \times W$. ($M_{s_video}$) is the final output of the whole attention module. Spatial attention mask is broadcasted over channel dimension of (F_c) during element-wise multiplication, outputting modified feature maps (F_s) with size $T \times C \times H \times W$. Equations of submodules can be written as follows

$$\begin{aligned} M_{c_video} &= a^c(F) \quad , \quad F_c = M_{c_video} \otimes F \\ M_{s_video} &= a^s(F_c) \quad , \quad F_s = M_{s_video} \otimes F_c \,, \end{aligned} \tag{3.1}$$

where $a^c()$ and $a^s()$ represent channel-temporal and spatio-temporal attention functions.

One way of going forward at this point is forwarding feature maps (F_s) to the classification part of the ResNet-18 model. However, as shown by Pérez-Rúa *et al.* [17] and observed in our experiments, success of models using this architecture is respectively low due to the vanishing gradient problem. Pérez-Rúa *et al.* [17] introduced two solutions in order to overcome this problem. We also adopted these solutions and integrated them into our model architecture.

The first solution aims to build a special pathway for attention modules' gradients to overcome the vanishing gradient problem. Since the model has four residual stages

and four attention modules, four spatial attention masks (M_{s_video}) will be calculated. At first, M_{s_video} generated by attention modules are processed with adaptive average pooling (AAP). AAP is designed to align all spatial dimensions to spatial size of feature maps obtained from the last residual stage. Spatially aligned masks are concatenated and forming a new mask (M_{meta}). This mask has size $N \times H \times W$ where N is the number of stages in the model, H $\times$ W is the spatial size of last stage's feature map.

A one layer 2D CNN model uses mask M_{meta} as input and outputs a feature refining mask (M_{frm}). The 2D CNN model uses 1×1 kernel to align channel size of M_{frm} with F_s^4. The mask M_{frm} is element-wise summed with final feature maps (F_s^4) produced before classification layers of the model to refine features. The refined feature map (F_{agm}) is used for classification. This solution method is called Attention Guided Feature Refinement (AGFR) and calculates (F_{agm}) as following

$$\begin{aligned} F_s^4 &= M_{s_video}^4 \otimes F_c^4 \,, \\ M_{frm} &= CNN(M_{meta}) \,, \\ F_{agm} &= F_s^4 + M_{frm} \,. \end{aligned} \tag{3.2}$$

The second solution requires training a student model that has the same architecture as a teacher model (see Figure 3.8). The teacher model has attention modules and AGFR. Also the teacher model used cross-entropy loss in the training process. On the other hand, loss function used to train the student model is different. It contains an additional regularization term to cross-entropy loss. This term is the distance between final feature maps of teacher and student model. It is calculated as follows

$$L_{mfgr} = ||F_{agm}^s - F_{agm}^t||_2 \,. \tag{3.3}$$

This solution method is called Mature Feature Guided Regularization (MFGR). The regularization term helps the model to improve predictive power by reaching a better local optimum [17]. The regularization term and cross-entropy is combined using a weight coefficient. The student loss is calculated as follows

$$
\begin{aligned}
L_{ce} &= -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i)\,, \\
L_s &= \alpha L_{ce} + (1-\alpha) L_{mfgr}\,,
\end{aligned}
\tag{3.4}
$$

where L_{ce} is cross-entropy loss, N is number of samples, y_i is label, $\hat{y}_i$ is predicted class, L_s is loss of the student model and α is the weight coefficient.

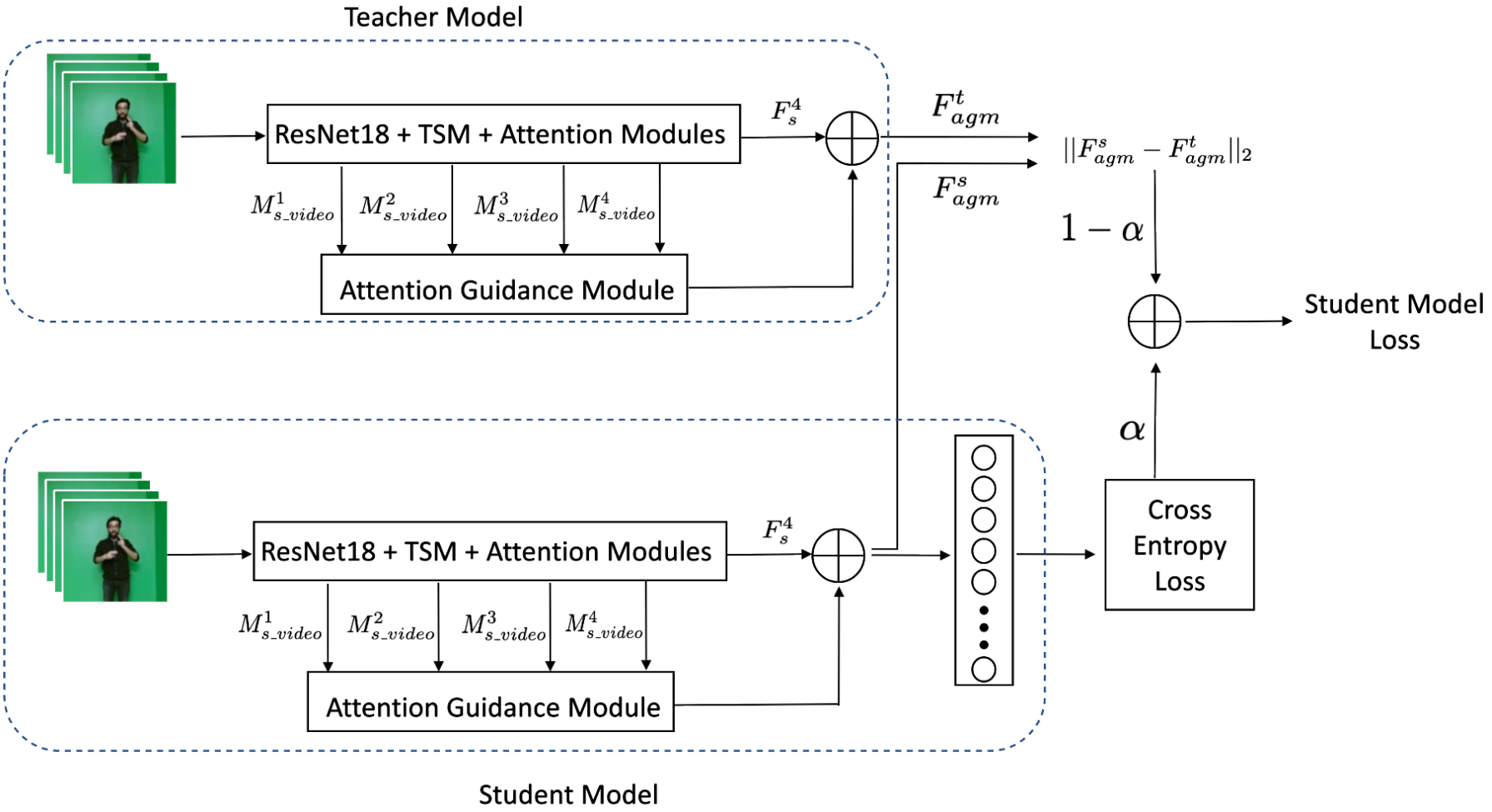

Figure 3.8. Illustration of teacher and student models' architecture in training process of student model.

4. EXPERIMENTS AND RESULTS

4.1. Dataset

The proposed model is trained and tested with Turkish isolated sign language dataset BosphorusSign22k [11]. The dataset contains 22,542 videos from 744 sign classes. The videos are from health, finance and commonly used sign glosses categories.

Glosses were performed by six native signers. Özdemir *et al.* [11] proposed using 18,018 videos performed by five signers to train models and 4,524 videos performed by the other signer to test models (see Figure 4.1). The introduced train-test split makes it a signer-independent dataset. We also applied this splitting strategy in our experiments.



Figure 4.1. Right most signer is used for testing, other ones are used for training.

All videos in the dataset were recorded with a Kinect v2 placed 1.5 meter away of signers that stood in front of a Chroma-Key background. The dataset has RGB videos, depth map, skeleton information and OpenPose [67] joint information. In this study, we used RGB videos and OpenPose data. RGB videos have 1920x1080 pixels resolution and 30 frames per second frame rate.

4.2. Data Preprocessing and Transformations

Certain preprocessing and transformation steps are applied to video frames in order to adapt data to model and improve the model's success (Figure 4.2). As the first step of data preprocessing, short edges of video frames are resized to 256 pixels. The aspect ratio is preserved during resizing operation. Since ResNet-18 pre-trained on ImageNet dataset is used as a backbone model, input images must satisfy certain conditions regarding spatial size. Resized video frames are cropped to size of 224×224 pixels in order to meet these conditions. Frames are cropped at random locations in the model training process. However, to evaluate different models under equal conditions, frames are cropped from the center in the testing process.

Moreover, the pre-trained model expects input values in a specified range. Therefore, values of video frames are scaled to [0-1] range, then normalized by using mean and standard deviation of ImageNet dataset. Video frames are also horizontally flipped randomly according to a given probability in the training process. If a video is decided to be horizontally flipped, then all frames selected from the video will be transformed. The flip probability is a hyper-parameter that needs to be fine-tuned. Horizontal flip is not applied during the testing process.

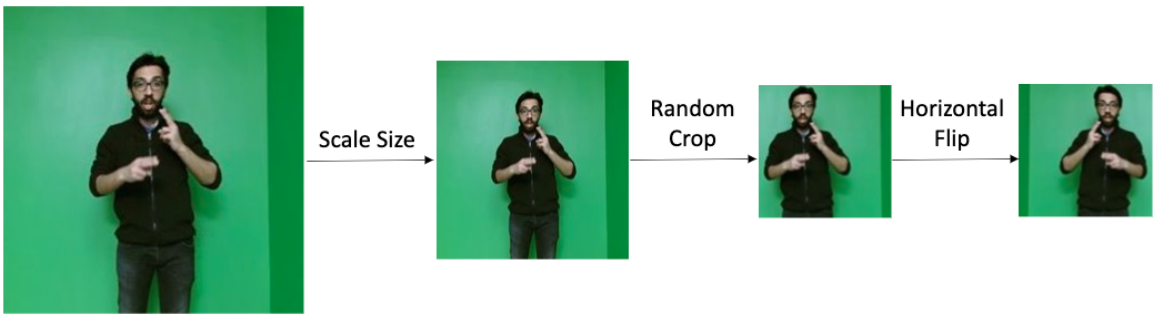

Figure 4.2. Illustration of the image preprocessing and transformation pipeline used in the training process.

4.3. Frame Selection

Video recognition models produced their final outputs according to predictions for selected frames from videos. Frames do not contain equally important information for the task. Some of them might be irrelevant or might contain redundant information [68]. Therefore, model success is directly related to which frames are selected from videos. Four frame selection methods are tried in this study (see Figure 4.3 for visualization):

(i) Linear Frame Selection: This method starts by selecting the first frame in the video. Then next ones are selected uniformly by skipping a number of frames. Number of frames to be skipped is decided according to the number of frames in the video and the number of frames that will be selected.

(ii) Segment Frame Selection: This method divides frames into multiple segments according to the given segment number. Then a frame from each segment is randomly selected.

(iii) Active Frame Selection According to OpenPose Data: The method aims to find frames where signers are actively performing some hand gestures. In this way, frames carrying more information about the sign gloss can be found. We named these frames as active frames. The method uses joint position data of signers in each video frame extracted with OpenPose pose estimation [67] in order to find active frames. More specifically, lunate bone positions in both hands, left and right hip positions together with neck position is used. Weighted ratio of hip and neck positions are used to determine a threshold position between them. If the signer's hand is above the threshold position in the frame then the frame is marked as active. Model's input frames are selected among these active frames in a temporal order.

(iv) Active Frame Selection According to MMPose Data: This method has the same logic with the previous one. The difference is the pose estimation algorithm used to find joint positions of signers. This method uses MMPose [69] instead of OpenPose.

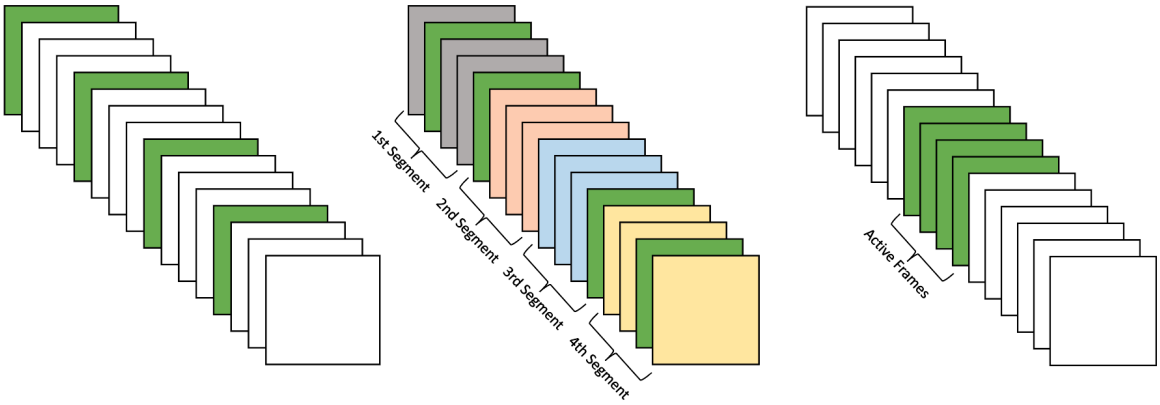

Figure 4.3. Illustration of different frame selection methods. From left to right: linear, segment and active frame selection. Green squares show selected 4 frames out of 16 in an example input video.

4.4. Temporal Shift Modules

Temporal shift modules (TSM) are inserted before the first CNN layer of each residual building block in the backbone model (Figure 3.4). The backbone network, ResNet-18, can perform temporal modeling thanks to TSMs. Since temporal modeling is crucial for video recognition, models with this ability should achieve better results than classic 2D CNNs.

We firstly compared the original ResNet-18 model and its TSM added version under the same conditions in order to see the effect of TSMs without attention modules and extra training procedures (AGFR and MFGR). Top-1 and Top-5 classification accuracy of models can be seen in Table 4.1. We can see that temporal modeling with TSM even in its basic settings greatly improves model accuracy.

After seeing the sole effect of TSM, we tested different approaches and settings to improve the model's classification accuracy. We tried image transformations, frame selection methods, using different numbers of video frames, batch sizes and model architectures.

Table 4.1. Comparison of ResNet-18 and ResNet-18 + TSM.

Model	Top-1 (%)	Top-5 (%)
ResNet-18	26.20	59.92
+ TSM	68.50	89.88

Horizontal flip is an image transformation technique to improve image and video classification models' accuracy scores by increasing generalization ability. We trained different models with changing flip probabilities. Results of the experiments can be seen in Table 4.2. Flip probability of 0.5 gives the best results in both Top-1 and Top-5 accuracy.

Table 4.2. Effects of using different horizontal flip probabilities tested with ResNet-18 + TSM.

Model	Flip Probability	Top-1 (%)	Top-5 (%)
ResNet-18 + TSM	0	54.75	79.22
ResNet-18 + TSM	0.3	66.97	89.83
ResNet-18 + TSM	0.5	68.50	89.87

Frame selection has a direct impact on a model's success as explained in Section 4.3. In this study, we applied linear frame selection, segment frame selection and active frame selection with both OpenPose and MMPose data. Linear frame selection assumes information is distributed uniformly in the video. While segment frame selection also has a similar logic, it adds some randomness and can increase the generalization power of a model. On the other hand, active frame selection focuses on informative and distinctive frames under assumption of frames with hand motions is more important for SLR. Experimental results presented in Table 4.3 are in line with previous explanations and the active frame selection with OpenPose achieves the highest classification accuracy.

Table 4.3. Effects of using different frame selection methods tested with ResNet-18 + TSM.

Model	Frame Selection	Top-1 (%)	Top-5 (%)
ResNet-18 + TSM	Linear	63.19	88.04
ResNet-18 + TSM	Segment	68.50	89.87
ResNet-18 + TSM	Active (OpenPose)	75.44	93.52
ResNet-18 + TSM	Active (MMPose)	71.35	90.67

SLR models produced their final sign class prediction by combining individual frame predictions. Therefore, selecting more frames from videos is expected to increase models' accuracy. On the other hand, using more frames increases computational cost of the model. Selecting the correct number of frames is important to find an optimum point for both model performance and efficiency. Performance of models trained with different numbers of frames is presented in Table 4.4. The SLR model gave the best results when 32 frames were used to train and predict. However, additional improvement of using 32 frames in accuracy is insignificant compared to its computational cost. Therefore, we decided to select 24 frames from sign videos.

Table 4.4. Effects of selecting different number of frames from videos tested with ResNet-18 + TSM.

Model	Number of Frames	Top-1 (%)	Top-5 (%)
ResNet-18 + TSM	8	66.64	89.43
ResNet-18 + TSM	16	75.44	93.52
ResNet-18 + TSM	24	82.62	96.55
ResNet-18 + TSM	32	82.63	96.59

When we analyzed train and test accuracies of TSM models, we saw that models are overfitting. We proposed two solutions to decrease overfitting and increase model performance with its ability to generalize. Firstly, we implemented two new models by adding dropout layers. The first of these models have dropout layers after its first three

building blocks with probability 0.3, 0.5 and 0.5, respectively. The second model has another dropout layer with probability 0.5 before its fully connected layer (see Figure 4.4 for model architectures). As for the second solution, we decreased batch size to improve accuracy because using large batches can cause to poor generalization [70]. Results of both approaches can be seen in Table 4.5. When batch size is 4, both models with dropout layers improved accuracy and adding another dropout layer had a positive effect in performance. Decreasing batch size to 2, improved accuracy of both models with and without dropout layers. Impact of decreasing batch size became more apparent in model without dropout layers. In this setting, the best performance is observed when TSM inserted ResNet-18 model is used with batch size of 2.

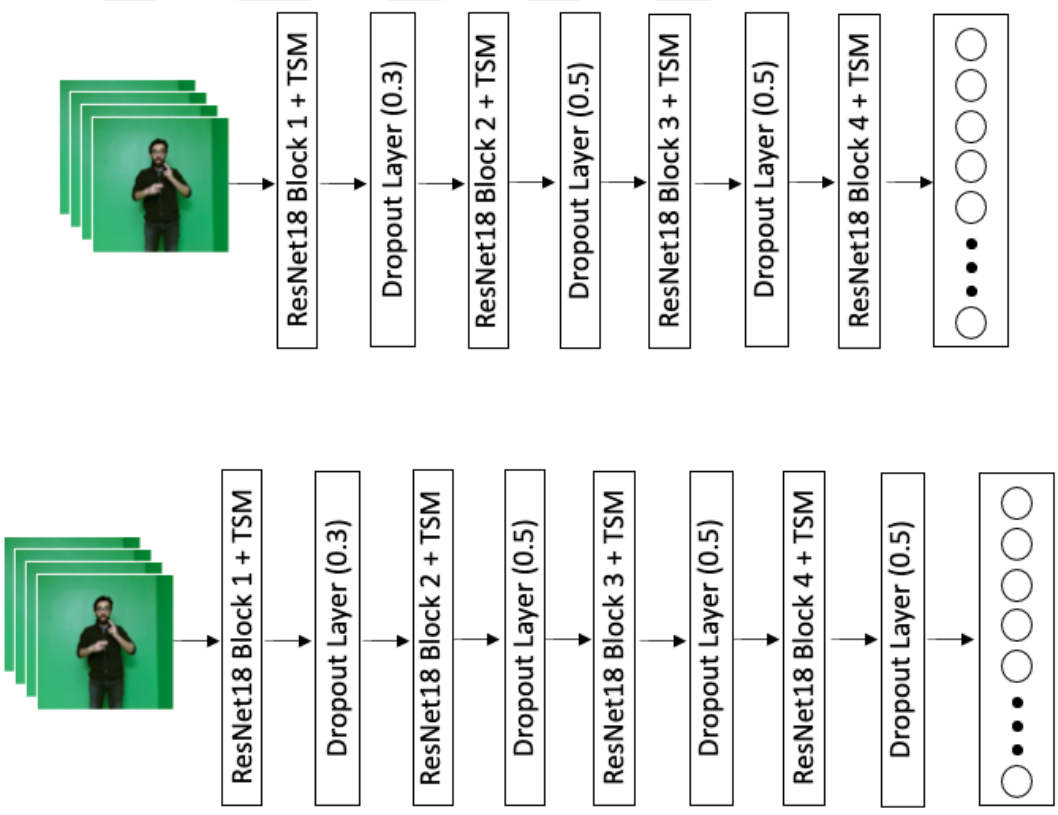

Figure 4.4. Visualization of the dropout layer inserted ResNet-18 + TSM architectures. From top to bottom: TSM with 3 dropout layers, TSM with 4 dropout layers.

Table 4.5. Comparison of selecting different model architectures and batch sizes.

Model	Batch Size	Top-1 (%)	Top-5 (%)
ResNet-18 + TSM	4	82.62	96.55
ResNet-18 + TSM	2	89.89	99.22
+ 3 Dropout layers	4	84.43	97.41
+ 4 Dropout layers	4	85.85	97.94
+ 4 Dropout layers	2	88.17	98.98

Firstly, we present results for adding TSM into a ResNet-18 model in Table 4.1. Then, we tested different hyper-parameters, frame selection methods, number of frames and model architectures. The best setting options according to performance become a guideline for attention model experiments. We set flip probability as 0.5, selected number of frames as 24 and model selection method as active frame with OpenPose.

4.5. Attention Modeling

Attention modules are added after each stage of the ResNet-18 model with TSM (Figure 3.5). The modules emphasize important and distinctive parts of extracted feature channels and suppress unnecessary parts. Training attention modules in the ResNet-18 backbone without additional arrangements limits it to reach its full potential due to the vanishing gradient problem as stated by Pérez-Rúa *et al.* [17].

In this section, we tested attention models with and without additional training procedures as explained in Section 3.2.3. Attention modeling reaches its top performance when both Attention Guided Feature Refinement (AGFR) and Mature Feature Guided Regularization (MFGR) is applied. In Table 4.6, we compare best results obtained with TSM models, their attention and AGFR added versions. While adding only attention modules decreases Top-1 accuracy by 1.48%, the best attention model with AGFR improves Top-1 accuracy by 1% compared to the best TSM model architecture. Moreover, AGFR improves Top-1 accuracy of attention model by 2.48%.

Table 4.6. Comparison of ResNet-18 + TSM with attention models.

Model	Top-1 (%)	Top-5 (%)
ResNet-18 + TSM	89.89	99.22
+ Attention	88.41	98.93
+ Attention + AGFR	90.89	99.13

4.5.1. Mature Feature Guided Regularization

Training models with applying MFGR has two main differences than other training processes. Firstly, we use a teacher model while training the student model. Secondly, a regularization term is added to cross-entropy loss to create loss function of the student model. We conducted experiments to try different options for both of these.

Loss function used in the student models' training has two main terms namely L_{ce} and L_{mfgr}. The two terms are multiplied with α and $1 - \alpha$. The α coefficient is a hyperparameter that needs to be optimized. If α value gets closer to 1, cross-entropy dominates loss functions. Both terms have equal effect on loss function when α is set to 0.50. Our experiments with different α values can be seen in Table 4.7. Best Top-1 accuracy is obtained when α is 0.90 so it is the value used in the next experiments.

Table 4.7. Effects of different α values in attention models with AGFR and MFGR.

Model	α	Top-1 (%)	Top-5 (%)
Attention + AGFR + MFGR	0.99	91.29	99.11
Attention + AGFR + MFGR	0.98	92.30	99.11
Attention + AGFR + MFGR	0.95	90.89	99.13
Attention + AGFR + MFGR	0.90	92.57	99.07
Attention + AGFR + MFGR	0.80	91.22	99.20
Attention + AGFR + MFGR	0.50	90.60	99.04

MFGR loss leads the student model to mimic feature maps of the teacher model in training process. MFGR introduced in study [17] uses a teacher model with attention modules and AGFR. Our best model so far is a student model shown in Table 4.7 with 92.57% Top-1 accuracy. This student model uses a regular teacher model as proposed in [17]. In this study, we proposed 2 new options to be used as a teacher model. Firstly, since the former student model has better classification accuracy, we used it as a teacher model in the training process of the new student model. Secondly, we tried using a teacher and a former student model as teacher models to train the new student model. As it can be seen from the results in Table 4.8, using a former student model as teacher model gave the best result with 92.97% Top-1 and 99.35% Top-5 accuracy.

Table 4.8. Effects of using different teacher model architectures in attention models with AGFR and MFGR.

Model	Teacher Model	Top-1 (%)	Top-5 (%)
Attention + AGFR + MFGR	Teacher	92.57	99.07
Attention + AGFR + MFGR	Student	92.97	99.35
Attention + AGFR + MFGR	Teacher + Student	90.71	98.65

4.6. Comparison With Other Studies

Different isolated sign language recognition models trained and published their results using BosphorusSign22k dataset. We compared our best result with these studies [3, 11, 13] in Table 4.9.

The proposed model with 92.97% Top-1 and 99.35% Top-5 accuracy gave better results than introduced solutions used Temporal Accumulative Features (TAF) [3], Mixed Convolution (MC) with 3D-2D ResNet models [11] and Improved Dense Trajectories (IDT) [11]. However, the proposed architecture using 3 MC models with spatio-temporal sampling and weighted score fusion [13] remains state-of-the-art with its 94.94% Top-1 and 99.76% Top-5 accuracy.

Table 4.9. Comparison of our model with other studies used BosphorusSign22k dataset.

Model	Top-1 (%)	Top-5 (%)
TAF [3]	81.37	97.47
MC3 [11]	78.85	94.76
IDT (HOG + HOF + MBH) [11]	88.53	-
MC3 + Spatio-Temporal Sampling + Score Fusion [13]	94.94	99.76
Proposed Model	92.97	99.35

We compared our proposed model with the state-of-the-art in Table 4.10. Even though our model didn't improve state-of-the-art in terms of classification accuracy, it has advantages in training process, number of parameters and memory need. Our proposed architecture uses one model to predict gloss of the sign. On the other hand, state-of-the-art fuses outputs of three models to predict the gloss. Our model has approximately 55% less parameters than the state-of-the-art. Furthermore, the proposed solution needs only full-frame RGB video frames and OpenPose data while the compared model requires cropped right hand, left hand, face images in addition to RGB video frames and OpenPose data. When both solutions were trained and tested by using only full frame RGB and OpenPose data, our model improves state-of-the-art by 6.06% in Top-1 and 1.18% in Top-5 accuracy.

In addition to performance comparisons with other studies, we checked inference duration of our best model. In order mimic conditions in a system without powerful GPU to some extent, we measured inference time of the model while running it on a Intel i7 CPU with 2.30 GHz. Average inference time for a video is measured as 0.567 seconds.

Table 4.10. Comparison of the proposed model with the state-of-the-art.

	Proposed Model	Gokce v.d. [13]
Backbone Network	2D CNN	3D CNN
Number of Predictive Models	1	3
Used Data	RGB Video Frames OpenPose Data	RGB Video Frames Right and Left Hand Crop Face Crop OpenPose Data
Number of Parameters	15.79 Million	35.07 Million
Performance with Same Data	92.97% / 99.35%	86.91% / 98.17%

4.7. Attention Visualization

We visualized spatio-temporal attention regions at different moments of a sign video (see Figure 4.5). At the beginning, the model attends both hands. However, motion of the left hand shifts the model's focus on that hand. When the signer raises her hands up to her mouth, the model mainly attends that region. When the signer starts to lower her hands, the model focuses on both of them. With completion of the sign, model attention shows similar characteristics to the beginning of the video.

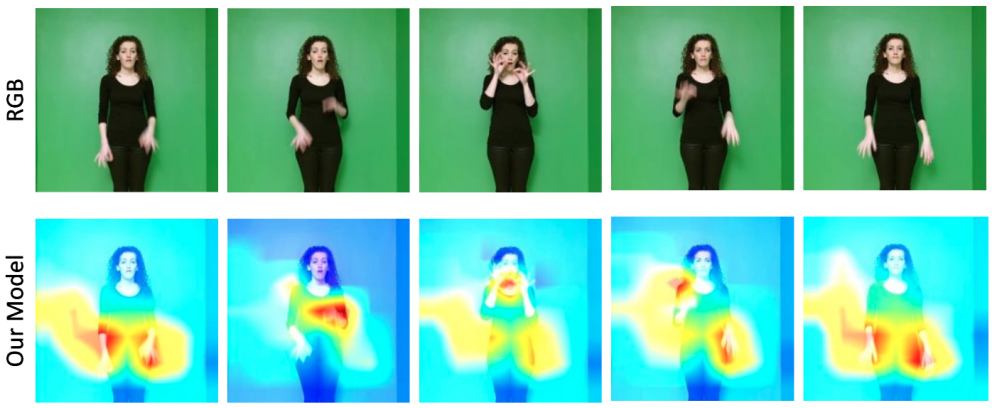

Figure 4.5. Visualization of spatio-temporal attention regions.

5. CONCLUSION

Deaf communities use sign languages to communicate with each other. The communication is performed through multiple visual cues performed simultaneously or in sequence. Therefore proposed models in the sign language recognition field should be able to learn certain motions in specific body parts together with relations between each other.

In this thesis, we worked on the Turkish isolated SLR problem and developed an attention model with temporal shift. The model aims to focus on specific parts of information extracted from sign video to predict glosses in BosphorusSign22k dataset. The dataset has 744 different sign glosses from health, finance and commonly used glosses categories.

The proposed approach has different contributions in the field than the solution introduced by Gökçe *et al.* [13]. Our architecture uses ResNet-18, a 2D CNN model, as its backbone model instead of a 3D CNN in order to reduce required memory and computational cost. Our study shows that TSM integrated 2D CNN models can be an alternative to 3D CNN for temporal modeling in SLR.

Since signs are performed through hand gestures, upper body movements and facial expressions, explicitly using them is helpful to increase model success. However, preparing crop images of these body parts from RGB video frames requires additional preprocessing. Moreover, it decreases reusability of proposed architecture with different datasets and problems. Our model uses channel-temporal attention and spatio-temporal attention to focus on important movements, locations and moments in the video frame while suppressing irrelevant ones. In this way, the model can give its attention to hands, upper body and face whenever they are informative without requiring additional data.

Results obtained in our experiments allowed us to make some inferences. Firstly, 2D CNN models with temporal shift and attention modules can give competitive scores in the SLR domain. Secondly, usage of 2D CNN as backbone model and solving temporal modeling issues without adding extra parameters to architecture keeps number of parameters, required memory and computational cost low. Moreover, achieving 92.97% Top-1 and 99.35% Top-5 accuracy by using only full frame RGB and OpenPose data increases the proposed model's adaptability to other datasets and to similar problems.

As future work, several improvements can be made in the model architecture. Firstly, the backbone model ResNet-18 could be substituted with deeper ResNet models. They are more generalizable and have lower training error than ResNet-18 [7]. Secondly, current architecture always selects 24 frames from videos. Designing an architecture that can dynamically adjust how many frames it should select according to video properties can be helpful to increase performance. Thirdly, changing how TSM works could improve model success. In this study, each TSM shifts the same ratio of feature channels in every depth of the network. However, modeling capacity is likely to change at different depths of the network [71]. Therefore, implementing a new TSM that shifts channels in different ratios according to depth of the residual block can increase model accuracy.

REFERENCES

1. Zhou, H., W. Zhou, Y. Zhou and H. Li, "Spatial-Temporal Multi-Cue Network for Sign Language Recognition and Translation", *IEEE Transactions on Multimedia*, Vol. 24, pp. 768–779, 2022.

2. Adaloglou, N., T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. J. Xydopoulos, K. Atzakas, D. Papazachariou and P. Daras, "A Comprehensive Study on Deep Learning-Based Methods for Sign Language Recognition", *IEEE Transactions on Multimedia*, Vol. 24, pp. 1750–1762, 2022.

3. Kindiroglu, A. A., O. Özdemir and L. Akarun, "Temporal Accumulative Features for Sign Language Recognition", *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 1288–1297, 2019.

4. Kadous, M. W., "Machine Recognition of Auslan Signs Using PowerGloves: Towards Large-Lexicon Recognition of Sign Language", *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*, Vol. 165, pp. 165–174, 1996.

5. Liwicki, S. and M. Everingham, "Automatic Recognition of Fingerspelled Words in British Sign Language", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 50–57, 2009.

6. Starner, T., J. Weaver and A. Pentland, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, pp. 1371–1375, 1998.

7. He, K., X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

8. Krizhevsky, A., I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Communications of the ACM*, Vol. 60, pp. 84 – 90, 2012.

9. Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

10. Simonyan, K. and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos", *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 1, pp. 568–576, MIT Press, Cambridge, MA, USA, 2014.

11. Özdemir, O., A. A. Kındıroğlu, N. Cihan Camgoz and L. Akarun, "Bosphorus-Sign22k Sign Language Recognition Dataset", *Proceedings of the LREC 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, pp. 181–188, 2020.

12. Tran, D., H. Wang, L. Torresani, J. Ray, Y. LeCun and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition", *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

13. Gökçe, Ç., O. Özdemir, A. A. Kındıroğlu and L. Akarun, "Score-Level Multi Cue Fusion for Sign Language Recognition", *Proceedings of the European Conference on Computer Vision Workshop (ECCVW)*, pp. 294–309, Springer-Verlag, Berlin, Heidelberg, 2020.

14. Vázquez-Enríquez, M., J. L. Alba-Castro, L. Docío-Fernández and E. Rodríguez-Banga, "Isolated Sign Language Recognition with Multi-Scale Spatial-Temporal Graph Convolutional Networks", *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3457–3466, 2021.

15. Koller, O., S. Zargaran, H. Ney and R. Bowden, "Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via Hybrid CNN-HMMs", *International Journal of Computer Vision*, Vol. 126, No. 12, pp. 1311–1325, 2018.

16. Camgoz, N. C., S. Hadfield, O. Koller and R. Bowden, "SubUNets: End-to-End Hand Shape and Continuous Sign Language Recognition", *IEEE International Conference on Computer Vision (ICCV)*, pp. 3075–3084, 2017.

17. Pérez-Rúa, J.-M., B. Martínez, X. Zhu, A. Toisoul, V. Escorcia and T. Xiang, "Knowing What, Where and When to Look: Efficient Video Action Modeling with Attention", *arXiv:2004.01278*, 2020.

18. Zhu, Y., X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha and M. Li, "A Comprehensive Study of Deep Video Action Recognition", *arXiv:2012.06567*, 2020.

19. Bobick, A. and J. Davis, "Real-Time Recognition of Activity Using Temporal Templates", *Proceedings Third IEEE Workshop on Applications of Computer Vision*, pp. 39–42, 1996.

20. Bobick, A. and J. Davis, "An Appearance-Based Representation of Action", *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 1, pp. 307–312, 1996.

21. Laptev, I., M. Marszalek, C. Schmid and B. Rozenfeld, "Learning Realistic Human Actions from Movies", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

22. Wang, H., A. Kläser, C. Schmid and C.-L. Liu, "Action Recognition by Dense Trajectories", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3169–3176, 2011.

23. Wang, H. and C. Schmid, "Action Recognition with Improved Trajectories", *IEEE*

International Conference on Computer Vision, pp. 3551–3558, 2013.

24. Sánchez, J., F. Perronnin, T. Mensink and J. J. Verbeek, "Image Classification with the Fisher Vector: Theory and Practice", *International Journal of Computer Vision*, Vol. 105, pp. 222–245, 2013.

25. Krizhevsky, A., I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", F. Pereira, C. Burges, L. Bottou and K. Weinberger (Editors), *Advances in Neural Information Processing Systems*, Vol. 25, pp. 1106–1114, 2012.

26. Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision (IJCV)*, Vol. 115, No. 3, pp. 211–252, 2015.

27. Ng, J. Y.-H., M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga and G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4702, 2015.

28. Wang, L., Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang and L. Van Gool, "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition", B. Leibe, J. Matas, N. Sebe and M. Welling (Editors), *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 20–36, Springer International Publishing, 2016.

29. Tran, D., L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks", *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, 2015.

30. Carreira, J. and A. Zisserman, "Quo Vadis, Action Recognition? A New Model

and the Kinetics Dataset", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017.

31. Lin, J., C. Gan and S. Han, "TSM: Temporal Shift Module for Efficient Video Understanding", *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7082–7092, 2019.

32. Kuehne, H., H. Jhuang, E. Garrote, T. Poggio and T. Serre, "HMDB: A Large Video Database for Human Motion Recognition", *International Conference on Computer Vision*, pp. 2556–2563, 2011.

33. Soomro, K., A. R. Zamir and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild", *arXiv:1212.0402*, 2012.

34. Abu-El-Haija, S., N. Kothari, J. Lee, A. P. Natsev, G. Toderici, B. Varadarajan and S. Vijayanarasimhan, "YouTube-8M: A Large-Scale Video Classification Benchmark", *arXiv:1609.08675*, 2016.

35. Kay, W., J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, A. Natsev, M. Suleyman and A. Zisserman, "The Kinetics Human Action Video Dataset", *arXiv:1705.06950*, 2017.

36. Goyal, R., S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax and R. Memisevic, "The "Something Something" Video Database for Learning and Evaluating Visual Common Sense", *IEEE International Conference on Computer Vision (ICCV)*, pp. 5843–5851, 2017.

37. Rastgoo, R., K. Kiani and S. Escalera, "Sign Language Recognition: A Deep Survey", *Expert Systems with Applications*, Vol. 164, p. 113794, 2021.

38. Vogler, C. and D. Metaxas, "Adapting Hidden Markov Models for ASL Recognition by Using Three-Dimensional Computer Vision Methods", *IEEE International*

Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Vol. 1, pp. 156–161, 1997.

39. Lee, C.-S., Z. Bien, G.-T. Park, W. Jang, J.-S. Kim and S.-K. Kim, "Real-Time Recognition System of Korean Sign Language Based on Elementary Components", *Proceedings of 6th International Fuzzy Systems Conference*, Vol. 3, pp. 1463–1468, 1997.

40. Koller, O., "Quantitative Survey of the State of the Art in Sign Language Recognition", *arXiv:2008.09918*, 2020.

41. Zhang, L.-G., Y. Chen, G. Fang, X. Chen and W. Gao, "A Vision-Based Sign Language Recognition System Using Tied-Mixture Density HMM", *Proceedings of the 6th International Conference on Multimodal Interfaces*, pp. 198–204, Association for Computing Machinery, New York, NY, USA, 2004.

42. Wong, S.-F. and R. Cipolla, "Real-Time Adaptive Hand Motion Recognition Using a Sparse Bayesian Classifier", *Proceedings of the International Conference on Computer Vision in Human-Computer Interaction*, pp. 170–179, Springer-Verlag, Berlin, Heidelberg, 2005.

43. Escalera, S., X. Baró, J. Gonzàlez, M. Á. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton and I. Guyon, "ChaLearn Looking at People Challenge: Dataset and Results", *Proceedings of the European Conference on Computer Vision Workshop (ECCVW)*, pp. 459–473, Springer International Publishing, 2014.

44. Camgöz, N. C., A. A. Kindiroglu and L. Akarun, "Sign Language Recognition for Assisting the Deaf in Hospitals", *International Workshop on Human Behavior Understanding*, pp. 89–101, 2016.

45. Özdemir, O., N. C. Camgöz and L. Akarun, "Isolated Sign Language Recognition

Using Improved Dense Trajectories", *24th Signal Processing and Communication Application Conference (SIU)*, pp. 1961–1964, 2016.

46. Camgoz, N. C., S. Hadfield, O. Koller and R. Bowden, "Using Convolutional 3D Neural Networks for User-Independent Continuous Gesture Recognition", *23rd International Conference on Pattern Recognition (ICPR)*, pp. 49–54, 2016.

47. Neverova, N., C. Wolf, G. W. Taylor and F. Nebout, "Hand Segmentation with Structured Convolutional Learning", *Asian Conference on Computer Vision (ACCV)*, pp. 687–702, 2014.

48. Koller, O., O. Zargaran, H. Ney and R. Bowden, "Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition", *Proceedings of the British Machine Vision Conference*, pp. 1–12, 2016.

49. Forster, J., C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. H. Piater and H. Ney, "RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus", *International Conference on Language Resources and Evaluation (LREC)*, pp. 3785–3789, 2012.

50. Von Agris, U., M. Knorr and K.-F. Kraiss, "The Significance of Facial Features for Automatic Sign Language Recognition", *8th IEEE International Conference on Automatic Face Gesture Recognition*, pp. 1–6, 2008.

51. Koller, O., J. Forster and H. Ney, "Continuous Sign Language Recognition: Towards Large Vocabulary Statistical Recognition Systems Handling Multiple Signers", *Computer Vision and Image Understanding*, Vol. 141, pp. 108–125, 2015.

52. Koller, O., N. C. Camgoz, H. Ney and R. Bowden, "Weakly Supervised Learning with Multi-Stream CNN-LSTM-HMMs to Discover Sequential Parallelism in Sign Language Videos", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 42, No. 9, pp. 2306–2320, 2020.

53. Vaezi Joze, H. and O. Koller, "MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language", *The British Machine Vision Conference (BMVC)*, pp. 1–16, 2019.

54. Wei, C., W. Zhou, J. Pu and H. Li, "Deep Grammatical Multi-Classifier for Continuous Sign Language Recognition", *IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pp. 435–442, 2019.

55. Neidle, C., A. Thangali and S. Sclaroff, "Challenges in Development of the American Sign Language Lexicon Video Dataset (ASLLVD) Corpus", *Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, Language Resources and Evaluation Conference (LREC)*, pp. 143–150, 2012.

56. Zhang, J., W. Zhou, C. Xie, J. Pu and H. Li, "Chinese Sign Language Recognition with Adaptive HMM", *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2016.

57. Huang, J., W. Zhou, Q. Zhang, H. Li and W. Li, "Video-Based Sign Language Recognition without Temporal Segmentation", *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pp. 2257–2264, AAAI Press, 2018.

58. Ji, S., W. Xu, M. Yang and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 1, pp. 221–231, 2013.

59. Köpüklü, O., N. Kose, A. Gunduz and G. Rigoll, "Resource Efficient 3D Convolutional Neural Networks", *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 1910–1919, 2019.

60. Hu, J., L. Shen and G. Sun, "Squeeze-and-Excitation Networks", *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

61. Woo, S., J. Park, J.-Y. Lee and I.-S. Kweon, "CBAM: Convolutional Block Attention Module", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, Springer International Publishing, 2018.

62. Wang, F., M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang and X. Tang, "Residual Attention Network for Image Classification", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6450–6458, 2017.

63. Dhingra, N. and A. Kunz, "Res3ATN - Deep 3D Residual Attention Network for Hand Gesture Recognition in Videos", *International Conference on 3D Vision (3DV)*, pp. 491–501, 2019.

64. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is All You Need", *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, 2017.

65. Wang, X., R. Girshick, A. Gupta and K. He, "Non-Local Neural Networks", *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

66. Zhou, B., A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Learning Deep Features for Discriminative Localization", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016.

67. Cao, Z., G. Hidalgo, T. Simon, S. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 01, pp. 172–186, 2021.

68. Fan, H., Z. Xu, L. Zhu, C. Yan, J. Ge and Y. Yang, "Watching a Small Portion could be as Good as Watching All: Towards Efficient Video Classification",

Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, pp. 705–711, 2018.

69. Contributors, M., "OpenMMLab Pose Estimation Toolbox and Benchmark", `https://github.com/open-mmlab/mmpose`, 2020, accessed on June 2022.

70. Keskar, N. S., D. Mudigere, J. Nocedal, M. Smelyanskiy and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima", *arXiv:1609.04836*, 2016.

71. Fan, L., S. Buch, G. Wang, R. Cao, Y. Zhu, J. C. Niebles and L. Fei-Fei, "RubiksNet: Learnable 3D-Shift for Efficient Video Action Recognition", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 505–521, 2020.

APPENDIX A: USAGE OF FIGURES

Figures prepared within the scope of this thesis work and whose copyrights were transferred to the publishing house were used in the thesis book in accordance with the "publishing policy for the reuse of the text and graphics produced by the author" on the website of the publisher.