HIROSHIMA UNIVERSITY

DOCTORAL THESIS

Spatio Temporal Features and its Possible Extensions for Action Recognition: from Handcrafted to Deep Learning

---

Author:

Novanto YUDISTIRA

Supervisor:

Professor Takio KURITA

Pattern Recognition Laboratory

Information Engineering

April 2, 2018

Declaration of Authorship

I, Novanto YUDISTIRA, declare that this thesis titled, "Spatio Temporal Features and its Possible Extensions for Action Recognition: from Handcrafted to Deep Learning" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Start a huge, foolish project, like Noah... makes absolutely no difference what people think of you."

Jalaluddin Rumi

HIROSHIMA UNIVERSITY

Abstract

Engineering

Information Engineering

Doctor of Philosophy

Spatio Temporal Features and its Possible Extensions for Action Recognition: from Handcrafted to Deep Learning

by Novanto YUDISTIRA

Action Recognition contains information over space and time because action possibly occurs in arbitrary positions, various scales and temporal dynamics which led to the need of robust yet low computational cost features. The progress of action recognition or video classification as broader topic has largely progressed given abundance of common datasets. However, there are still rooms to improve recent features in which still questionable ranging from handcrafted to learned features such as spatio temporal auto correlation, multi layered wavelet packet, motion superpixel localization, and mixture expert via deep Convolutional Neural Network (CNN). Results show that it is either improving state of the art or computationally efficient compared to the existing features.

Acknowledgements

I would like to express gratitude to Professor Takio Kurita as academic supervisor for long life support of my PhD study and researches, his patence, helps, motivations and knowledges. His guidance helps me through tough research experiences and papers submissions. His supervision has significant role in the PhD success.

I also would like to express my appreciation to thesis committee of Professor Takio Kurita for efforts, manuscript review, and valuable assistantances. Associate Professor Toru Tamaki who review the thesis. And also wish to thank to JASSO (Japan Student Services Organization) and Soroptimist for financial helps during doctoral period. The special service, hospitality, and assistance are also devoted to Gradueate School of Engineering, Integrated Science and Art building, and Departement of Engineering of Hiroshima university staffs.

I also express gratitute to my parents who always supporting me mentally and financially, my wife for her patience and support, my little daughter who always cherish me, and my friends who cannot be described one by one for supporting and accompany me from first semeseter until now.

And for very utmost, i always feel grateful to god, Allah SWT because without His permission in this life, this thesis would never finish.

Contents

List of Figures

List of Tables

List of Abbreviations

HOF	**H**istogran of **F**lows
HOG	**H**istogram of **G**radients
HLAC	**H**igher order **L**ocal **A**uto**C**orrelation
CLAC	**C**ubic **L**ocal **A**uto**C**orrelation
GLAC	**G**radient **L**ocal **A**uto**C**orrelation
FLAC	**F**lows **L**ocal **A**uto**C**orrelation
PF	**P**acket **F**lows
WPT	**W**avelet **P**acket **T**ree
SVM	**S**upport **V**ector **M**achine
LOOCV	**L**eave **O**ne **O**ut **C**ross **V**alidation
BoF	**B**ag **o**f **F**low
HSV	**H**ue **S**aturation **V**alue
RGB	**R**ed **G**reen **B**lue
SEEDS	**S**uperpixels **E**xtracted via **E**nergy **D**riven **S**ampling
CNN	**C**onvolutional **N**eural **N**etwork
VGG	**V**isual **G**eomtry **G**lroup
ResNet	**R**esidual **N**et

Chapter 1

Introduction

We are facing the era where informations are spreading easily through web. It turns to be beneficial for video understanding due to the abundance of data (big data) provided by users. As the videos are growing in numbers, it is required to classify generally based on scenes or human actions. In this research, human action classification is considered as topic to be enhanced and explored. Action recognition has been tremendous active area of research with many methods have been proposed. However, several issues need to be tackled by recent feature extraction either handcrafted or deep learned. Because the video has high computational complexity due to its dimension size, it is compulsory to find features that have good trade off between speed and accuracy. There are several unexplored properties by accommodating autocorrelation between frames inside spatio temporal space, making use of temporal dynamics using wavelet approach, precise motion localization via superpixel, and mixture of expert to blend spatial and temporal stream of Convolutional Neural Network (CNN).

1.1 Local Autocorrelation

The progress of action recognition become more advanced in terms of computation time and accuracy to recognise. Sadanand et al. (Sadanand and Corso, 2012) proposes Action Bank that uses orientation filters along space

and time but this is rather computationally expensive and its applicability is questionable by the fact that it has such computational complexities to be run under recent machines. To describe the movement along action cycle, we use optical flow in which its densities can be categorized as dense or sparse optical flows. In terms of dense optical flows, recently researchers avoid optical flows because of wild, non-regular properties and the presence of camera jittering etc which turns into low performance. We employ the edge based motion to be adopted in our action recognition framework. Edges or its residuals (Kim et al., 2010)(Sundberg et al., 2011) have ability to specify the movement objects. We use Canny edge to obtain edge response of all over frames. Edge plays significant role to suppress the flows that is not part of foreground or object interest such as human. By suppressing using Canny edge detector, noise of motion over entire flow field is minimized. These method potentially supports motion compensation which has been an issue to be solved such as (Jain, Jegou, and Bouthemy, 2013). It is robust to some extent to the presence of camera motions, yet it does not explicitly handle the camera motion. In most cases, this will be effective to distinguish the impact of camera movement and independent actions. We also introduce the motion and vector autocorrelation over time properties that we will consider through **??**.

To realize local autocorrelation derived from motion models, we consider to utilize spatial binning in the form of histogram of oriented flows (HOF). Under this framework, we can possibly apply action recognition in real time cases such as (Matsukawa and Kurita, 2010). As evaluation, KTH action dataset (Schuldt, Laptev, and Caputo, 2004) is presented and classification scheme of crossvalidation using linear SVM is employed.

As the first contribution of this paper, to the best of our knowledge, we are the first to evaluate both efficiency and classification performance of optical flow and local vector autocorrelation for action recognition over time. It is called flow based local autocorrelation over time (FLAC over time) and find it

achieving high recognition rates without loss in speed compared to the state of the arts. We also consider to propose method that is robust and has low computational cost utilizing multiresolution of flow fields. Most importantly, it obtains some informations which are not captured by existing features. We bring an encoding technique known as vector autocorrelation of flows to the field of action recognition.

Recent works have been conducted with both autocorrelation and optical flows. There are researchers have proposed either utilising dense optical flows such as (Jain, Jegou, and Bouthemy, 2013)(Wang et al., 2011) or autocorrelation (Matsukawa and Kurita, 2010) for action recognition. One can utilise the information of every pixel to obtain a dense correspondence, or merely use sparse feature points (Liu, Yuen, and Torralba, 2016). Basically, sparse techniques only need to process some pixels from the whole image while dense techniques process all the pixels or windows. For real time applications, Lucas-Kanade's sparse optical flows (Lucas and Kanade, 1981) accuracy might be enough since dense optical flows are relatively slower but the latter has useful advantage to gain more accurate result than the former one. The most popular of a dense optical flow algorithm is Gunner Farneback's Optical Flow (Farnebäck, 2003). Optical flows, despite of its wilderness motion, has rich information about movement that useful if it is treated properly. Specifically for dense optical flows, it has characteristic that is useful for sampling larger area rather than sparse optical flows. A rather different view is adopted in (Wang et al., 2011) where the decomposition of motions is represented at the trajectory level. In this work, the sequence of motion forms flow field sequence, however exploiting these trajectories could be the future issues of our proposed method.

We use flow fields to gather features information rather than pixel-wise of image frames that has been widely employed. In other parts, action has duration over time or cycles. In pixel level, (Schindler and Van Gool, 2008) has

explained about how many frames suitable for recognizes. By using some HOG and scale invariant features (SIFT) descriptor (Lowe, 1999). It is revealed that up 10 frames, 1- 7 frames are enough to capture the action. For our case, in flow fields level, 5 - 15 flow fields enough to capture snippet of actions. Trade of between computation and accuracy for large video dataset become more remarkable for recognition benchmark. Very short cycle will cause loss information about actions. Rather than using detection and body tracking that is not reliable and computationally burden in realistic human action recognition on video, for features derived from pixel-wise, (Shi, Petriu, and Laganiere, 2013) has made observation on random sampling strategies using local parts models. For optical flows, (Ke, Sukthankar, and Hebert, 2007) decides how many variation of box sampling there inside video from optical flows. On other hand, we use multiresolution window and grid sampling over the flow field. It turns out that by using non overlapping sliding windows is enough to produce comparable performance.

1.2 Deep Wavelet Packet

Intelligent vision system (Otsu and Kurita, 1988) especially action recognition is growing topics in computer vision and pattern recognition. It is gaining its popularity since Shultz work which also provides well-known dataset (Schuldt, Laptev, and Caputo, 2004). Correspondingly, there are many real-world recognition applications that exploit human actions such as surveillance camera, video classification, sports analysis, human-computer interaction etc which its application becomes more demanding as the hardware quality became more sophisticated. It leads action recognition to be challenging problems since human performs in many ways and camera can take object in a various manner. For instance, in appearance aspect there are many kind and color of clothes are attached to human. Occlusion is also another

problem that sometimes distracts real motion into false or less informative motion. From camera aspect, various scales of human object are captured because of distance matter. Moreover, camera can be static or dynamic which is also emerging problem that remains wide open. From human aspect, action with its variability of speed, background, clothes, illumination is dynamic. To tackle this problem, handcrafted HOF itself cannot be used to describe the variability of dense optical flows. Thus, it is reasonable to make extension to form sequence of HOF and make some sort of decompositions to discover general and distinctive pattern. High level is required to give semantic meaning to classes. These issues lead to many feature representations proposed by researchers to discriminate action types performed by humans.

The focus of recognition should be concentrated more into feature representations, especially for action recognition. There are many previous works that proposed various features whether it is spatiotemporal, template-based, high level or medium level. Many approaches have been proposed as action representation can be categorized into interest point (Chakraborty et al., 2012) (Klaser, Marszałek, and Schmid, 2008) ,slow features (Sun et al., 2014)(Theriault, Thome, and Cord, 2013)(Legenstein, Wilbert, and Wiskott, 2010), motion (Fathi and Mori, 2008), high level convolution (Sadanand and Corso, 2012), and shape and appearance based (Lin, Jiang, and Davis, 2009). Interest point based needs detection and description step in which the detection phase plays significant role to find most salient representation of actions. Its collection of points are sparse enough to be featured but if the detection fails to produce suitable representation due to occlusion or noise, it turns to weaken the performance. High level and medium level convolution holistically change the low-level features into more sophisticated measurement by means convolution (Sadanand and Corso, 2012). Under spatial and time space, it would be computationally burden. Even though it is biologically inspired features, there are many other approaches that are comparable to

the convolution based algorithm which has lower complexity. In this paper, we would like to focus on motion-based features to time-varying motion distribution to make use temporal dynamics by means Haar Wavelet Packet decomposition.

Flow, despite its drawback against noise, has some advantages. It has lower complexity than convolution based thus it would be advantageous if the flow is densely sampled rather than sparsely such as (Wang et al., 2011)(Uijlings et al., 2014). In terms of drawbacks, optical flows leave the problem of occlusion or camera movement that distract motion from true human motion. Moreover, dense sampling has advantage of smoothness that can handle fast motion. In challenging dataset such as KTH where the small jittering on camera occur, human silhouettes appearance and low resolution will reduce the accuracy of capturing human action cycle given spatiotemporal space. However, for more challenging dataset such as UCF Sports where the object of interest may appear at different angles relative to camera and frame to frame change is not smooth, motion-based features produce weak performance result. Thus, various temporal dynamics within action can be captured if more detail motion is decomposed and hopefully robust to aforementioned noises.

Many improvements (Sun et al., 2014)(Byrne, 2015)(Lan, Wang, and Mori, 2011)(Wang et al., 2011)(Jain, Jegou, and Bouthemy, 2013)(Matsukawa and Kurita, 2010)(Ke, Sukthankar, and Hebert, 2007)(Yu, Sommer, and Daniilidis, 2003) have been made utilizing local motion as base representation model. Motions are various over time especially if execution time is dynamics. We collect HOFs temporally and enrich each bin as the temporal channel. Specifically, from collected video frames, for every local channel, multi-resolution histogram is extracted based on Haar Wavelet packet in specified depth. More precisely, given all of the video, we learn how to decompose based on the high pass or low pass signal.

The use of tracking and template based leaves drawback that is high computation which in turns difficult for real-time system. BoF is more robust to extent template based and tracking in terms of noise and background changing. More importantly, it preserves discriminative information of local geometric structure of features. The parameter of BoF must be tuned to obtain optimal class specific codebook. In spatial term, different resolution local window used for extracting histogram also influence the codebook generation. However, it is computationally expensive to sample various resolution sizes. In temporal term, how long the cycle to be considered is taken into account for forming the features structure, however, translation invariant vectors are required since action occurs in arbitrary frames.

In this section, Deep Local Wavelet Packet Histogram of flow (Packet Flow) is derived from dense optical flows. Optical flow is generated from pixel movement from one frame to next frame along a sequence of time-varying image intensities. Packet Flow has three-fold contribution as

(i) the introduction of Deep Local Wavelet Packet Histogram for each bin which temporally integrate histograms over a sufficiently long time period under noise and occlusion in several levels,
(ii) the analysis of Wavelet Packet depth of motion histogram derived from sequence of flow field which able to reduce spatial noise and extract detail of temporal dynamics,
(iii) the framework of efficient features computation which using generalized intersection kernel of SVM classifier and its potential extensions.

Dataset is selected based on assumption that there is no preprocessing such as detection and tracking. Thus, there is no guarantee of clean optical flows between frames and action of each sample is in constant movement. Moreover, video is captured at high rate and various time interval.

Temporal information has been paid attention in action recognition with local motion pattern as focus of interest. Some improvements have been progressed along this path. Motion has been an interesting representation to be considered as it is intuitively suitable for object moment characterization. In real case, camera motion sometimes encounters and become the main obstacle for generation or selection of true human optical flow. Especially for dense optical flow such as (Farnebäck, 2003)(Fleet and Jepson, 1990), optical flow points are presented in grid form wherever motion presents.

There are issues regarding dense optical flow utilization which are discriminating independent actions or eliminating camera motion. To tackle the problem, some researchers have adopted motion compensation to reduce unwanted optical flow due to camera jittering and unrelated actions such as separating dominant and residual motion without recovering 3D motion (Jain, Jegou, and Bouthemy, 2013). After such compensation, some features are used as descriptor. (Jain, Jegou, and Bouthemy, 2013) has proposed features derived from kinematic properties. Another researcher has proposed motion boundary features as descriptor of dense flow (Wang et al., 2011). After compensation, by densely sampling with the step size of 5, feature points are gathered. Dense sampling will cover the entire frame as much as possible depending on the step size. However, it is considered as computationally expensive because the greater step size it takes, it would burden time complexity. There is approach to compensating and extracting the features of motion trajectories but the analysis of temporal information in space and time is yet to be explored (Jain, Jegou, and Bouthemy, 2013). They also lose the valuable information provided by local flow field of different temporal dynamics.

There are various application using derivation of temporal information along image sequence of which in pixel level such as grouping detection by means game theory (Oshin, Gilbert, and Bowden, 2014) and complex event by means temporal dynamics (Bhattacharya et al., 2014). Rather than pixel

wise and gradient wise, we consider HOF originated from flow fields as a base of our measurement. In practice, we can adapt spatial binning by dividing it into subregions spatio-temporally. This way, can produce invariant capability of being shifted.

Number of cycles or number of frames need to be decided for construction of local geometric information and it is still open question in action recognition case (Shi, Petriu, and Laganiere, 2013)(Schindler and Van Gool, 2008). Some other researches have done in pixel space or higher level order. It is needed to construct best spatio temporal features that optimally represent local features and also cover intra class variability. In the previous work Yudistira and Kurita, 2015, we used autocorrelation similar to (Kobayashi and Otsu, 2008) in pixel level to capture similarity given spatio temporal space. Even though it shares similarity in terms of properties, it leaves chances to be extended since the base representation itself is HOF.

We use bag of features to accommodate spatio temporal geometric information. Specifically, class specific codebook generation is adopted to extract intra class variational clusters which has advantage for discrimination. This way can minimize noise without lose information about local features compared to pooling. Most of reference, extend its features using BoF method which has many advantages and applicabilities to be adopted in video (Somasundaram et al., 2014)(Oshin, Gilbert, and Bowden, 2014)(Wang et al., 2011). Rather than accumulation or concatenation of local spatio temporal features, it delivers sparse representation that has been well known to produce high performance in classification scheme. BoF also promises efficient dimension number of representation to accommodate low complexity for recognition while also preserve local geometrical features information. Moreover, in terms of video recognition, it preserves temporal information especially to capture dynamic motion along space and time.

To define motion dynamics, recent use of motion extraction is done by

means hand-crafted features such as HOG, HOF and SIFT (Lowe, 1999) (Liu, Yuen, and Torralba, 2016) machine learned such as Slow Features Sun et al., 2014(Theriault, Thome, and Cord, 2013), relative motions (Oshin, Gilbert, and Bowden, 2014), multi-level representation (Wang, Qiao, and Tang, 2016), rank pooling (Fernando et al., 2017) or Long-short Term motion (Lan et al., 2015). Trajectories are popular extension of hand-crafted features to define motion (Chen and Zhang, 2016) (Wang et al., 2011) but lack information about various temporal dynamics. Slow Features try to adopt the principle of slowness. In natural scene, change of time scale varies if continuous slow varying motions are obtained from quick varying motion, it would bring underlying sensory input of brain to gather information about motion. However, it is computationally expensive since need effort for unsupervised step to extract slow features. It is termed to be more powerful in handling noisy motion than Integrated Subspace (Le et al., 2011). There is also proposed method which employs various length of block size inside video called Long-short Term motion Lan et al., 2015 but it is prone to noise. Rather than those, starting from handcrafted features, we propose to extend sequence of HOF and then interpolate along temporal to be decomposed using Haar Wavelet Packet. This will bring richer information of motion in form of multi-resolution because there are various signal packets either in high pass or low pass. Similar works have been done in another topics such as image texture (Hadjidemetriou, Grossberg, and Nayar, 2004) (Laine and Fan, 1993) and signal processing (Lee and Shin, 2000)(Gokhale and Khanduja, 2010). This approach is easy to be implemented and has low complexity to be analyzed in many levels of detail.

1.3 Motion Superpixel

Motions, despite its potential drawbacks in terms of camera jittering, noise, and occlusion, there is an advantage to explore its communal behaviour as one of the main sources of activity movements over entire video frames. To this end, superpixels along motion sequences that contain rich time time-series and geometrical information can be utilized as source of motion informations. In this study, we used the superpixel approach to segment motion into structured flow fields. Unlike conven- tional superpixels that segment a region by using pixel informations, in this proposed method, motion-based (angular and magnitude) superpixels are independently constructed at each frame. Unlike pixel-based frames, flow field is a movement of one pixel to the next frame by its angular and magnitude space which is estimated using an optical flow algorithm. While superpixels are well established in pixel-level segmentation, we introduce superpixel segmentation using the superpixels extracted via energy-driven sampling (SEEDS) algorithm to discriminate flow field comprising of flow vectors along the directions of motion. The contribution of this research is two-fold: firstly, we apply the concept of motion superpixels and their time evolution to the field of video classification along with its possible extensions like wavelet decomposition and secondly, produce evaluation results that demonstrate the usefulness of this method in producing results comparable to the state of the arts.

SEEDS was proposed by Bergh et al. as a texture imaging algorithm useful for various object recognition (Bergh et al., 2012). Following this, video SEEDS was introduced as a method for tracking superpixel continuity through time (Bergh et al., 2013). Energy driven superpixels produced by growing segmented regions that iterate have been shown to be computationally efficient and robust. One challenge in using superpixels in video is determining how to accurately track the actual evolution and endpoint of a

superpixel. The examination of this problem suggest the possibility of adopting superpixels in activity recognition tasks, particularly those in flow space. Figure 1 shows how tracking from one superpixel to its respective superpixel over time behaves. It should be tracked by following average direction of flows inside one superpixel or selecting the nearest position of consecutive superpixel. The consideration about nearest position is because in nature flows of two consecutive flow fields do not largely change thus constructed superpixel does not significantly move. Although motion features are usually represented by optical flows, there are possible extensions, including locality and the use of higher order local autocorrelation (Shiraki et al., 2006) based on analysis of autocorrelation between spatially and temporally neighbouring pixels. The viability of such approaches hinges on whether it is possible to adapt correlations in flow fields and the advantages of doing this in recognition exercises. The spatio temporal dynamic of activity cannot be neglected in feature construction. Unfortunately, differential operator multi scale analysis is prone to losing low frequency information. Recently developed and well founded methods such as spatiotemporal interest points (STIP) (Laptev, 2005), dense trajectories (Wang et al., 2011), and scale-invariant feature transform (SIFT) (Lowe, 1999) are prone to bias at coarse scales while learned features like convolutional neural network (CNN) (Sermanet et al., 2013) are prone to overfit if training data is not much. To better understand spatial and temporal dynamics or time series properties at high or low frequencies, we propose the use of wavelet packet decomposition.

1.4 Mixture of Expert via Gating CNN

The video classification task has become an interesting topic in computer vision and pattern recognition because of its dynamic scenes and objects, which vary either spatially or temporally, making it challenging to design suitable

and robust handcrafted features. The evolution of convolutional neural networks (CNNs) has led to significant changes in the way features are being learned. For instance, convolutional filters process pixels considering many aspects such as neighboring pixels and the shapes they form. Therefore, deep CNNs produce many parameters, which is advantageous for the classification task, especially for the classification of video. However, a CNN still needs gating to determine which modality should have more weight than the others. For instance, the gating network should be able to a spatial stream's output more heavily than a temporal one if spatial cues are more salient than motion cues, and vice versa.

Video classification using CNN has achieved significant improvement since the use of a collection of still images and ImageNet weights to be fine tuned on two stream network. In this paper, we implemented the two-stream CNN proposed by Simonyan (Simonyan and Zisserman, 2014a) for human action recognition, which uses spatial and motion streams using the Chainer framework (Tokui et al., 2015). Space and motion basically complement each other in nature to characterize activity in videos. There is evidence that integrating RGB channels and optical flow as a representation of space and motion respectively overcomes severe overfitting while increasing testing accuracy (Simonyan and Zisserman, 2014a)(Feichtenhofer, Pinz, and Zisserman, 2016)(Park et al., 2016). However, how to weight each spatial and motion feature remains an open question.

A feature weighting mechanism is required to find the optimal solution given a set of solutions. Using a gating scheme enables a network to be better trained to understand under what conditions the weights of the RGB part should be increased and under what conditions the optical flow should be weighted more heavily. Despite its advantages, there is one drawback of running gating scheme; it requires a large amount of CPU/GPU memory because of, in the case of bi-modalities, a large architecture of three networks

(two expert networks and one gating network). In this research, each expert network is trained independently and the gating network is then trained to weight each modality before integration.

The gating network scheme is primarily the same as the mixture expert scheme. It is basically inspired by the associative cortex of the brain, which can handle information integration from many sources. Based on (Stein, Stanford, and Rowland, 2009), it is evident that the presence of the associative cortex is needed to improve the perception of the environment by the brain. This conclusion is drawn from a study of cats with a deactivated cortico-collicular system, where it was found that the ability to integrate target neurons in the superior colliculus is disrupted. Correspondingly, our gating CNN scheme follows the natural cortico-colliculus to improve perceptions. Therefore, the main contribution of the gating CNN scheme is to select local patterns that best describes a decision. Because of the high number of degrees of freedom of scenes inside videos, spatial information alone is not enough to describe the target classification, which is sometimes disrupted from one scene to another. Information from one source might be not enough for a CNN to classify the video, regardless of millions of parameters, which tend to lead to overfitting. There are three possibilities to overcoming this problem: adding a larger variety of inputs, increasing training data, or gaining help from another source. When multisource information is considered as input, normalization is required to make their spaces comparable. For instance, if all frames from one modality are at fixed scales, another source such as motion must be at a fixed scale of the same size to enable the network to perform better with respect to perception. Whenever the output of softmax cross-entropy is retained from each expert stream, the gating network's output weights both experts' output (the output dimensions of the gating network are two when only two expert networks are used).

The success of CNNs has led to a new trend in activity recognition research. Video activity recognition is basically formed by a set of images for which CNNs have demonstrated superior classification. Recently, large image datasets such as ImageNet have been used to enrich the network with the aim of improving the accuracy of image-based classification tasks. However, the incorporation of other sources of information is needed to further improve perceptual accuracy. (Simonyan and Zisserman, 2014a) proposed a two-stream CNN that use spatial and temporal cues and performs simple fusion by averaging and using a support vector machine (SVM). Moreover, (Wang et al., 2016) improved the method of training the two streams by segmental sampling and used predefined fixed weights for the final feature fusion. Many fusion methods have been proposed, for instance, late fusion using a loss function (Feichtenhofer, Pinz, and Zisserman, 2016) or feature amplification-multiplication (Park et al., 2016). However, we assume that independent streams and loss are more natural because each stream has more freedom to learn depending on its specific task. This motivates us to propose an independent gating CNN architecture.

To summarize, the main contributions of this work are as follows: 1) We propose a framework for a gating scheme that is more accurate than if we use only one expert network or merely predefine fixed weights for many expert network outputs. 2. We propose our method using two deep models: expert and gating networks with independent loss functions and adaptively weighted outputs of every sample.

Previous studies based on still images have significantly contributed to human activity recognition, such as the two-stream CNN approach used by Simonyan et al. (Simonyan and Zisserman, 2014a), who proposed a very deep network for image recognition (Simonyan and Zisserman, 2014b). Their proposed method was extended to a temporal segment network (Wang et

al., 2016), which segments the whole video sequence and trains each segment based on its respective network, achieving higher accuracy. However, how to fuse or integrate all streams is still an open question. Before deeply learned features became popular, there were many research approaches to video classification using various methods, especially handcrafted methods such as spatiotemporal features (Somasundaram et al., 2014), dense trajectories (Wang et al., 2011), and local autocorrelation (Yudistira and Kurita, 2015). Three-dimensional (3D) CNN was the first attempt to train spatiotemporal features for video classification using deep CNNs. However, it had an overfitting problem due to the lack of available training videos (Karpathy et al., 2014). Later, a YouTube video dataset was provided and late fusion and early fusion for 3D CNN were introduced. Slow features can be learned using deep learning, which is advantageous for action recognition (Sun et al., 2014), however, the effectiveness of deep learning over handcrafted systems is still not evident. A breakthrough was proposed with a two-stream network that uses spatial and motion streams and fuses them by simple averaging and SVM fusion. Furthermore, it gains complementary information, which in turn improves accuracy. This approach adopts transfer learning from the large-scale ImageNet dataset and inherits the characteristics of image classification for video action recognition. Time series information was considered by (Hochreiter and Schmidhuber, 1997)(Gers, Schmidhuber, and Cummins, 1999) in a long short-term memory network, which is basically a gated version of a recurrent neural network.

A multiplicative gating scheme has been introduced by previous researchers for object detection, language modeling, people re-identification (Ahmed, Jones, and Marks, 2015), or video classification. Gated object detection was introduced by Xingyu at al. (Zeng et al., 2016) to make use of visual cues of different scales and resolutions. A gated CNN for language modeling was

presented by Yann et al. (Dauphin et al., 2016), who proposed a gating mechanism that outperforms long short-term memory-based gating. There is one multiplicative gating scheme for video classification (Park et al., 2016). It introduces feature amplification to perform soft gating on intermediate feature maps, which is a different approach to our work, which uses an additional gating network instead. Recently, weighted image segmentation for scene geometry and semantics has been an issue in deep learning applications (Kendall, Gal, and Cipolla, 2017). If we consider adding one gating network for weighting, it is necessary to calibrate measurements because the gating network itself is for predicting uncertainties. We consider how to manually define learning rate parameters to stabilize expert networks. How to provide an adaptive learning rate such as ESGD (Dauphin et al., 2016) remains an open issue. A natural gating network is able to learn non-linearities such as natural transformations (Hadsell, Chopra, and LeCun, 2006) for weighting the expert streams. (Feichtenhofer, Pinz, and Zisserman, 2016) proposed a fusion scheme for both RGB and optic flow streams in various layer position and trained it as a model using one loss function. Our approach is different from this in that we use a separate loss for the RGB, flow, and gating streams, which are independently trained in a sequential way. The gating output is trained to weigh both the last layer of the RGB and flow before fusion and classification.

1.5 Organization of Disertation

This thesis is organized as follows:

In Chapter 2, we develop autocorrelation of consecutive frames along temporal of flow fields. It is proposed by assumption that motion has correlation given various timestamps. It is shown comparable accuracy and speed on KTH dataset.

In Chapter 3, spatio temporal space of video is dynamic, thus, it is better to explore multiresolution decomposition via wavelet. Results shows that deeper feature resolution gaining additional information. However, despite of that, we have to select the decomposition band and level which is possibly done during experiment. The interesting property of wavelet decomposition

In Chapter 4, motion superpixel is presented which basically localization based on optical flows to gather precise salient feature. The difficulty of extracting motion features is mainly camera motion distraction. To this end, motion compensation is used as pre processing step before superpixel extraction. Every motion superpixel is then feed to bag of features for the sake of sparseness.

In Chapter 5, gating CNN is proposed to weight each stream of network (spatial and motion). We propose gating CNN which is gating stream based on CNN. The difficulty of gating CNN train is as the expert networks are saturated, overfitting is occurred on test data, thus, gating CNN can not training "real" variation of output expert.

Chapter 2

Multiresolution of Local Autocorrelation

We propose method for fast action recognition and comparable performance using local autocorrelation of optical flows over time. To capture action movement, dense optical flows is generated along sequence of video. Optical flows sometimes yield noise of motions that distract object of interest from another object motions and background. We suppress this by using edge based optical flow. The HOF vector is extracted from each window resolution and correlate its consecutive flow fields within cycle using local autocorrelation over time. It will gather richer information from movement while also gaining discriminative features than standard histogram methods. Comparison shows that the comparable performance is achieved over state of the arts.

2.1 Multiresolution

We quantize HOF of flow fields into 10 flow orientations and divide spatial flow fields into five resolutions ($X \times Y$: 15×15, 20×20, 25×25, 30×30, 35×35) which is multiresolution (Fig. **??**)). The size of spatial resolution may varies depending on the dimension of video. We decide spatial binning of every resolution, in which the size of window could be various to form

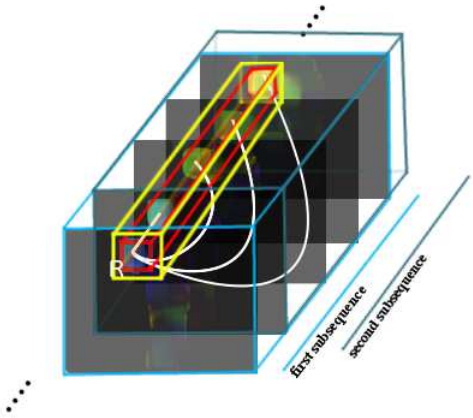


FIGURE 2.1: Multiresolution flow field autocorrelation over time.

histogram along consecutive flows. It also possible to adapt scale invariant features over flow field because the object interest may appear in many scales. Even though the spatial striding windows are not overlapping, but for the sake of computational speed, we can show that we still can achieve comparable accuracy. Even if denser sampling over flow field intuitively yield higher accuracy, we show that with spatial grid sampling and certain subsequence of consecutive flows is enough to capture actions.

2.2 Local vector autocorrelation

We present a causal action recognition method which uses only information from a collection of subsequences (snippets) within full sequence (video) of flow fields as figured out in Fig. **??**). Dense optical flows are densely extracted from local edges and capture its motion sequence over time inside a snippet. Autocorrelation is calculated over cycle and motion channels to learn how correlation between one flow field and another within subsequence of cycle is. Autocorrelation of temporal related motion path is generated as integration of these subsequences.

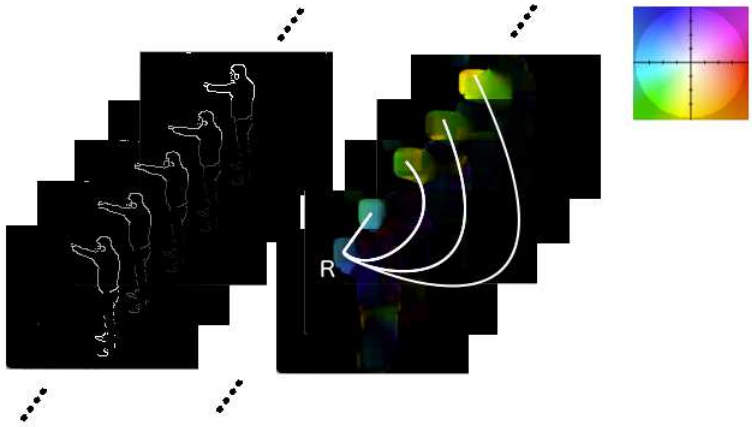


FIGURE 2.2: Autocorrelation over time across sequence of flow fields.

Kobayashi et al. (Kobayashi and Otsu, 2008) has published method for image recognition using vector autocorrelation of intensities gradient named gradient local autocorrelation (GLAC) and normal local autocorrelation (NLAC) which related to higher order local autocorrelations (HLAC) (Otsu and Kurita, 1988) and cubic higher local autocorrelations (CHLAC) (Matsukawa and Kurita, 2010). Different from HLAC and CHLAC that correlates pixel intensities and GLAC that calculate gradient of intensities, we proposed to use flow field. Rather than intensity gradient vector, we use HOF vector and sequentially correlate the flow field over time. HOF forms vector that later could be correlated over cycle. The shift-invariant features that is the nature characteristic of autocorrelation can be naturally applied to local descriptors such as as in SIFT (Lowe, 1999) or HOG by simply dividing regions into several subregions (spatial binning). Spatial binning reduces shift-invariance but increases discriminative power and this problem can be solved using integration of local autocorrelations. Even if the detection problem is considered, local autocorrelation needs not shift invariant property due to roughly aligned person images is compensated by shifting the detection window over the

subregions inside image. Although GLAC and NLAC are completely shift-invariant, thus for accuracy comparisons, the image region is divided into local regions or blocks such as 4×4 blocks, and the GLAC/NLAC features extracted over blocks are integrated into a final feature vector in the similar mean as as SIFT. In FLAC, similar approach is considered by using various resolution of blocks and performing integration for all over local regions within flow field.

We propose an efficient method to exploit local auto correlation information by vectorizing, stacking, summing and normalizing flow local autocorrelations over time, in which we name it the FLAC over time. The method was motivated by the result achieved by previous works on actions (Matsukawa and Kurita, 2010) that employ local autocorrelation. The key idea in our method is to exploit the computation of local autocorrelation into dense vector field rather than computing pixel-wise autocorrelation as the original CHLAC does. The use of zero-th order prone to be redundancy (Kobayashi and Otsu, 2008) which reduce the performance, thus, only local autocorrelation is considered. Local autocorrelation between two vectors is also necessary since the core information is revealed from alteration of object.

This consecutive flow fields reveal such repetitive pattern that can be captured using autocorrelation. It contains two kinds of vector correlations of flows histogram: spatial correlations derived from displacement spatial flow field and orientation vector correlations derived from the products of its the element values. We do not correlate flow themselves but HOF vectors which are quantized and represented sparsely. The order of auto-correlation is 1 which enables extraction of sufficient geometric characteristics together with local displacements a_i.

Let the S and s is sequence and subsequence with a number of flow fields contain inside it respectively. Let F be an flow field and $r = (x, y)^t$ be a position vector in F. Thus, we can formulate $s \in S$, $F \in s$, and $r \in F$. Each

flow can be represented in terms of the magnitude $m = \sqrt{x^2 + y^2}$ and the orientation angle $\theta = \arctan(x, y)$. The orientation θ is quantized into D orientation bins by voting weights of its magnitude to the nearest bins, and is described as a sparse vector f($\in R^D$), called the HOF vector.

There are two kind of correlations of flows that are correlations between reference (Fig 2.3a) with its consecutive vector (Fig. 2.3b) and orientation correlations derived from the products of the its element values (Fig. 2.3c). We do not correlate flows themselves but HOF vectors which are quantized and represented sparsely. Thus, the practical formulation of FLAC over time (A) is given by

$$A(R, a_t, t, j) = \sum_{s \in S} \sum_{r \in F} \sum_{0 \leq i \leq n(f(r))-1} f(r)[i] f(r, r + a_t, t)[i - j] \qquad (2.1)$$

(**??**) shows spatial correlations derived from displacement vector and time interval of which a_i and t respectively and orientation correlations derived from the products of the element values f. This is due to the empirical fact that, in HLAC, the auto-correlations of binary values, i.e., quantized data, are better for establishing recognition than those of the pixel values themselves. Where f , a_t, and t is vector of HOF, displacement mask in t^{th} flow field and t^{th} consecutive flow field relatives to the reference respectively. In practice, we can apply masking in the center of local region of reference flow field to calculate HOF (Fig. 3a). For its consecutive flow field given δt, center location that exactly the same with reference's center location and its neighboring locations are both employed (Fig. 3b) by the fact that motion between two consecutive flow field is not largely change. Because the displacement of flows given certain limit of sequence is not large, we can set a_i into low degree of number. In this work we set the value of a_i into 1. The displacement intervals also are the same in horizontal,vertical, and diagonal directions. We

TABLE 2.1: Accuracy results over four number of flow fields per subsequence

Resolutions	5 flow fields	10 flow fields	15 flow fields	20 flow fields
15x15	0.72 +/- 0.05	0.79 +/- 0.04	0.78 +/- 0.03	0.79 +/- 0.02
20x20	0.64 +/- 0.05	0.71 +/- 0.06	0.70 +/- 0.02	0.69 +/- 0.07
25x25	0.78 +/- 0.04	0.84 +/- 0.04	0.84 +/- 0.02	0.84 +/- 0.02
30x30	0.59 +/- 0.07	0.64 +/- 0.08	0.65 +/- 0.07	0.64 +/- 0.05
35x35	0.65 +/- 0.07	0.70 +/- 0.06	0.72 +/- 0.05	0.71 +/- 0.06
Combined	0.88 +/- 0.04	0.90 +/- 0.04	**0.91 +/- 0.03**	0.91 +/- 0.03

want to distinguish either the actor is going down, up, or diagonal within subsequence.

Autocorrelation between vectors can be applied using various lag. Where product between reference and its consecutive flow field is defined by discrete autocorrelation of signal at lag j. Lag means how much difference indices of both elements are to be correlated. Commonly, this way is implemented in speech extraction such as (Shannon and Paliwal, 2006). Total representation is given by concatenation of n pair between reference flow field and its counterpart given the values within T and J of $t \in T : 1 \leq t \leq n(s) - 1$ and $j \in J : -(n(f(r) - 1)) \leq j \leq n(f(r) - 1)$ respectively. Each local region accumulates its own autocorrelations which is then L2-normalized. In the case of calculating features in many subregions of an flow field, we can apply a method similar to the integral image approach by summing all over local windows of flow field, thus the total feature representation is of size $n(A) \times n(T) \times n(J)$.

2.3 Experiment

In this experiment, the extracted features are classified by using the linear SVM. The proposed methods were tested on the KTH dataset (Fig. **??**), details of which are in (Schuldt, Laptev, and Caputo, 2004). KTH consists 600 action videos and 6 classes of boxing, handclapping, handwaving, jogging,

running and walking. It contains 600 actions. Each class has 100 action videos represented by 25 different people under four different scenarios, which are outdoors, outdoors with different scales, outdoors with different clothes, different intensities and indoors with static and homogeneous background. Its resolution is 120 $\times$ 160 (*height* $\times$ *weight*) for all over videos in dataset.

FIGURE 2.3: Six different action classes of KTH dataset

We set our parameters of each part namely dense optical flow, resolution windows, orientation bins, number of subsequence interval, HOF weighting, displacement masks, and normalization method as follows:

[Dense optical flow] Flanerback optical flow parameter setting are 0.5, 3, 15, 3 $\times$ 3, 5, and 1.2 for pyramids scale, number of pyramid layer, averaging window size, number of iteration of each pyramid, size of pixel neighbourhood for each pixel to find polynomial expansion, and Gaussian standard deviation used for derivatives smoothing based on polynomial expansion respectively.

[Resolution] Resolution windows is the first processing step that may affect the final performance. We applied for KTH dataset 15x15, 20x20, 25x25, 30x30, and 35x35 resolutions as it is most effective, whereas the greater or less resolutions is not significant to influence performance result.

[Orientation bins] Orientation bins are evenly spaced over [0,360] degrees (signed flows). 10 orientation bins in 360 degrees is used in this experiment. (Kobayashi and Otsu, 2008) shows that finer binning increases performance while the signed gradient works better than the unsigned gradient. For autocorrelations of orientations, signed flows seem to be preferable.

[Subsequence interval] The parameter to consider length of action cycle is necessary. By giving T time within subsequence, it is closely related to the how many flow field per subsequence of full sequence video are to be processed. Intuitively, the greater number of flow fields can capture more reliable motion pattern of actions. This subsequence is often called action snippet.

[HOF Weighting] The weight of HOF is qualitatively defined as magnitude of flows, as it affects how strong the flow is respecting to its angular. This influence the quantisation results of HOF.

[Displacement mask] The displacement mask of consecutive flow fields relative to reference flow field can be varied. In this experiment we set into 1 for consideration that displacement interval to form motion between two consecutive flow fields is not large.

[Normalisation] We adopt L2 normalisation which refers to normalisation by L2-norm. These normalisation are applied to whole feature vector.

For comparison to the other methods, we compare overall performances of the proposed methods with those of the other previous methods that use the same KTH dataset. Note that we use stratified k-fold crossvalidation that produce number of testing data equal for all classes.

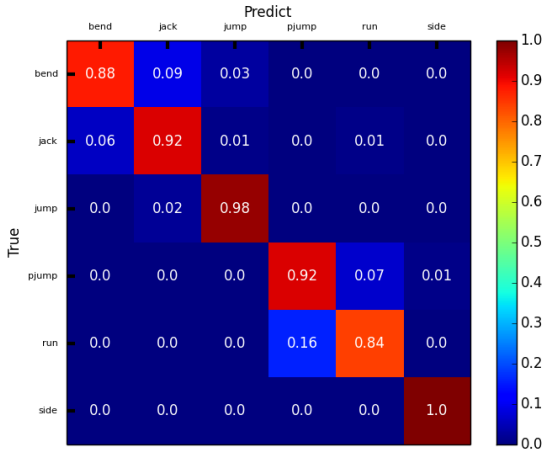


FIGURE 2.4: 10 fold crossvalidation of KTH by means LOOCV

2.4 Results

Table 1 presents our experimental results with 5, 10. 15, and 20 flow fields within subsequence over KTH dataset. The tests were run with the parameters setting that have been described before. For each flow fields per subsequence, we use 15x15, 20x20, 25x25, 30x30, 35x35 resolution windows or channels and classification is performed using linear SVM with cost parameter of 1. The mean accuracy and standard deviation are given by aforementioned stratified 10 fold crossvalidation. The best performance is revealed with 15 flow fields within a subsequence using combination of all the resolution windows. The performance almost always improves as the number flow fields within subsequence increase until such number of flow fields within subsequence. For the number of flow fields of 20 within a subsequence, it can be shown that it has given same accuracy mean as 15 but with larger deviation. It can be assumed that 5 - 15 number of flow fields within subsequence is enough to capture the action cycle. Note that standard deviation is very low for all results that means how effective and consistent the accuracy results are.

Table 2 compares our results to state of the arts. On KTH, we obtain 92.3

TABLE 2.2: Accuracy results over four number of flow fields per subsequence

Methods	Performance (KTH dataset)	Evaluations	fps	Frame size
(Somasundaram et al., 2014)	83.4%	train-test split	0.6	360x288
(Ke, Sukthankar, and Hebert, 2007)	80.9%	LOOCV	N/A	N/A
(Fathi and Mori, 2008)	90.5%	LOOCV	0.2-5	160x120
(Ta et al., 2010)	91.2%	LOOCV	0.5	160x120
(Mikolajczyk and Uemura, 2011)	95.3%	LOOCV	0.12-0.18	160x120
(Chakraborty et al., 2012)	96.3%	random 80:20 train:test	0.9	160x120
Ours	**92.3%**	LOOCV	**1.2-16**	160x120

% which is comparable to the state of the arts. Two of which are above and the rest below our result. As in Fig. **??**), the most confusing is between jogging and running class while many methods have been proposed face the same confusion classes problem. Even though such confusion has been revealed, result shows the capability of shift invariant from FLAC since the position of running, jogging, and walking can constantly change within sequence of frames. The action of jogging and running is quite similar motion characteristics that we confident these discrimination problem can be done by modifying parameter of our framework or by adopting bag of features.

In terms of speed performance, by using CPU of 3.7 GHz Quad-Core Intel Xeon E5, 12 GB 1866 MHz DDR3 ECC, and OSX platform for all extraction phases, we test on the video of which duration is 360 frames and resolution is 160x120. Because 15 flow fields per subsequence has been chosen as ideal cycle length, we use this as comparison to the other published methods. The running time of extraction depends on the machines, size of video, the size of resolution window and how many resolution windows to be utilized. The execution time we have achieved for whole sequence to process canny edge, dense optical flow, HOF and autocorrelation over time is between 1.2 to 16 fps (frames per second) which is comparable to the others. 16 fps speed can be realized if only 35x35 resolution windows is utilized. For combination of all aforementioned resolution windows, it would take 1.2 fps. As in table 2, there are many methods that use the same KTH dataset or another dataset as complexity testing which in turn leads to the multiform video size and also the machines that they used are different, we consider this the difference is not significant.

Chapter 3

Deep Wavelet Packet of Local Dense Optical Flows

Action recognition with dynamic actor and scene has been a tremendous research topic. Recently, spatio temporal features such as optical flows has been utilized to define motion represention over sequence of time. However, to increase accuracy, deep decomposition is necessary either to enrich information under location or time varying actions due to spatio temporal dynamics. To this end, we propose algorithm consists of vectors obtained by applying multi-resolution analysis of motion using Haar Wavelet Packet (HWP) over time. Its computation efficiency and robustness have led HWP to gain popularity in texture analysis but their applicability in motion analysis is yet to be explored. To extract representation, a sequence of bin of Histogram of Flow (HOF) is treated as signal channel. Deep decomposition is then applied by utilizing Wavelet Packet decomposition called Packet Flow to many levels. It allows us to represent action's motions with various speeds and ranges which focuses not only on HOF within one frame or one cuboid but also on the temporal sequence. HWP, however, has translation covariant property that is not efficient in performance because actions occur in arbitrary time and sampling's location is various. To gain translation invariant capability, we pool each respective coefficient of decomposition for each level. It is found that with proper packet selection, it gives comparable results on the KTH action and Hollywood dataset with train-test division without localization. Even if spatiotemporal cuboid sampling is not densely sampled

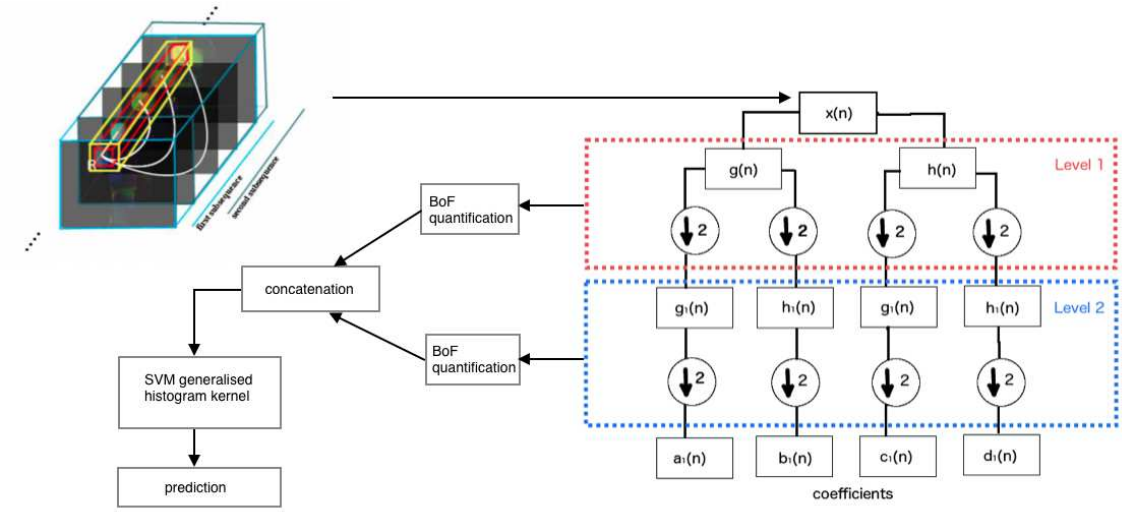


FIGURE 3.1: General architecture of proposed method.

like of baseline method, we achieve lower complexity and comparable performance on camera motion burdened dataset like UCF Sports that oftenly motion features such as HOF do not perform well.

3.1 Haar Wavelet Packet

Basically, Haar Wavelet is orthogonal and symmetry signal decomposition method giving fascinating characteristic for multi-resolution invariant. This is done recursively every level depending on the depths. To extract Packet Flows (PF) in spatiotemporal space, we propose a framework as in Fig. **??**. Flow fields are extracted along the frames and cuboid using pre-defined sizes and used to extract local HOF. To this end, in spatiotemporal space, the length of flow fields subsequence must be determined. The action appears in a various temporal sequence that has to be captured. Thus, a collection of HOF along subsequence of time (Fig. **??**) must be decided before deriving Haar Packet Wavelet using multi-resolution wave analysis. Frequency-based histogram for each bin is decomposed into several resolutions depending on vector size as elaborated in section **??**. Over temporal sequence, the output vector of every subspace is interpolated to gain the advantage of continuous

time series information while also reduce noise as described in section **??**. In our hypotheses :

1. Deeper decomposition can estimate various resolutions of sampled features.

2. Subspace must be selected to obtain good information.

3. Pooling is required to add spatiotemporal translation invariant capability in spatiotemporal space.

As explained in section **??**, to gather more spatiotemporal invariant, every subspace is normalized and then pooled in form of energy. Every level of decomposition is then quantized into the bag of features (BoF) for the sake of sparseness. Class specific BoF is used as sparsity transformation as plotted in section **??**. Multi-level BoFs are concatenated to be fed into the classifier. The suitable classifier sparse histogram vector is SVM with generalized histogram kernel in which described in section **??**.

3.1.1 Multi-resolution Haar Wavelet Packet and time-varying histogram on spatiotemporal space

We are considering each bin sequence of HOF sequence as time series vector and later is concatenated to be time series histogram. The basis of features is HOF to compute temporal dynamics of motion between consecutive flow field frames. The frequency of each bin is decomposed equally into separated groups with equally fixed bandwidth. As metrics, the multi-resolution scheme with Haar Wavelet Packet has been utilized along time interval.

By decomposing temporal resolution with Haar Wavelet Packet (HWP), besides enriching HOF bins, instances from the same class are varied to increase similar characteristics. Our proposed method directly implement existing HWP. In terms of formulation, suppose H is a representation of a sequence of HOFs with i is fixed number of bins, s is an index of the bin, n is a number of sequential HOF in a sequence, and j is an index of HOF inside the sequence. There is B which is a sequence of bins as part of H overtime giving timing varying n. Thus, $H = \{h_0, h_1, ...h_{n-1}\}$ where for every $h_j \in H$ has $b_s \in B$. Specifically, for each H there exist $B = \{b_0, b_1, ...b_{i-1}\}$. The each sequence of time-varying b is formulated by giving $B_s = \{h_0(s), h_1(s), ...h_{n-1}(s)\}$.

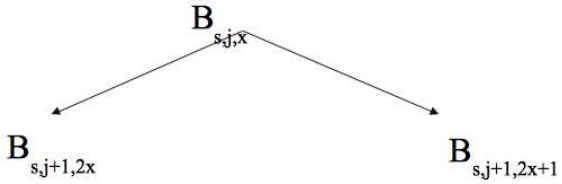


FIGURE 3.2: Wavelet Packet Tree decomposition based on initial orthogonal basis

As in Figure **??**, B_s is sequence and j is a depth number, x is node number. We can define each $B_{s,j+1,2x}$ ((**??**)) and $B_{s,j+1,2x+1}$((**??**)) as low pass and high pass function respectively. This refers to classical Haar Wavelet function in which utilized such that :

$$B_{2x}(i) = \sqrt{2} \sum_{k=0}^{2N-1} h(k) W_x(2i - k) \tag{3.1}$$

$$B_{2x+1}(i) = \sqrt{2} \sum_{k=0}^{2N-1} g(k) W_x(2i - k) \tag{3.2}$$

DWPT in time series at first level applies low and high level with :

$$N = 1, h(0) = h(1) = \frac{1}{2}$$

and

$$g(0) = -g(1) = \frac{1}{2}$$

where :

$$W_{2x}(i) = W_x(2i) + W_x(2i-1)$$

and

$$W_{2x+1}(i) = W_x(2i) - W_x(2i-1)$$

Enriching histogram by Haar Wavelet Packet has multi-resolution characteristics that are ability to capture large or small motion cycles by means frequent decomposition. Besides, more detail information of various motion is extracted. There is the psycho visual relation that human visual system is transmitting objects in the multi scale manner. Its computation efficiency and constructed wavelet bases have led to gaining popularity in texture analysis. However, suitable and appropriate wavelet bases depend on the purpose and can be considered only based on experiment. Meanwhile, the proposed idea is to decompose each dynamic bin quantization of histogram along temporal space. This is beneficial for capturing discontinuities in every bin. The general properties of spatiotemporal histogram intersection are explained in Figure **??**. The algorithm is estimated every predefined cuboid volume. We

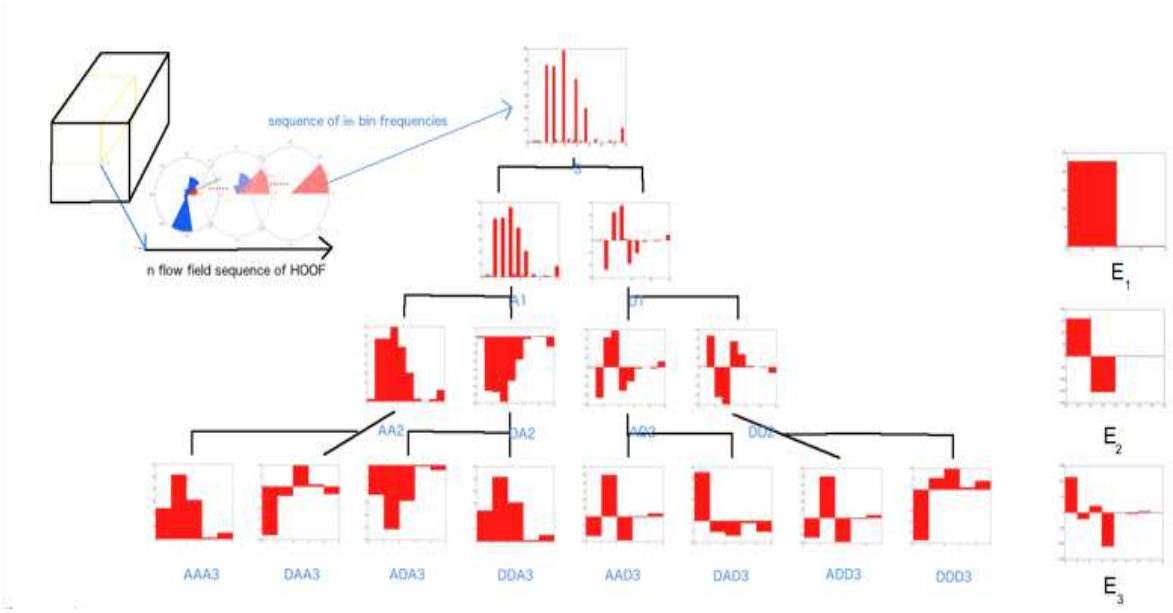


FIGURE 3.3: Decomposition of each bin of HOF given n flow fields. For each bin acts like channel in which frequency of flow fields are collected and decompose itself into many depths. Each depth of Wavelet Packet decomposition is pooled based on its packet.

can define the size of cuboid before undertaking HOF and its temporal sequence. Wavelet packet decomposition is performed in multi-level depending the length of the initial feature vector. The deeper wavelet packet level, the window size is getting narrower, dense, and more detail in scale. The deeper level has an advantage of a variety of temporal dynamic of motions. Because of natural characteristics of WPT, PF is simple and relatively fast to be implemented. The basic idea of the algorithm is decomposition of histogram sequence. Every Wavelet packet contains coefficient to justify the value of how dominant or distributed the waves are. It has a real-valued vector that spans across different levels and decompositions.

Classification performance for each level is investigated by various wavelet packets. All decomposition packets of every level are selected as vector representation and investigate its performance. For 40 sequence of motion bin and 3 depths, there are 4 levels (0,1,2,3). For every level, it constructs wavelet packets. We investigate every layer used for feature calculations.

1. Level 0 (S) is represented by sequence of HOF's frequencies from one bin channel.

2. Level 1 (A1, D1) is represented by 2 wavelet packets with dyadic high pass decomposition coefficients (A1) of 10 and low pass decomposition coefficients (D1) of 20 each. From then, energy values are calculated as sum of each packet E1.

3. Level 2 (AA2, DA2, AD2, DD2) is represented by 4 wavelet packets with dyadic high pass decomposition coefficients (AA2) and low pass decomposition coefficients (DA2) derived from level 1 high pass packet of 10 each and high pass decompositions (AD2) and low pass decomposition coefficients (DD2) derived from level 1 low pass packet of 10 each. From then, energy values are calculated as concatenation of sum of each packet E2.

4. Level 3 (AAA3, DAA3, ADA3, DDA3, AAD3, DAD3,ADD3, DDD3) is represented by 8 wavelet packets with dyadic high pass decomposition coefficients (AAA3) and low pass decomposition coefficients derived from level 2 high pass filter of level 1 high pass filter, high pass decomposition coefficients (DAA3) and low pass decomposition coefficients derived from level 2 low pass filter of level 1 high pass filter, high pass decomposition coefficients (ADA3) and low pass decomposition coefficients (DDA3) derived from level 2 low pass filter of level 1 high pass filter, high pass decomposition coefficients (AAD3) and low pass decomposition coefficients (DAD3) derived from level 2 high pass filter of level 1 low pass filter, high pass decomposition coefficients (ADD3) and low pass decomposition coefficients (DDD3) derived from level 2 low pass filter of level 1 low pass filter, all of which has 5 coefficients. From then, energy values are calculated as a concatenation of

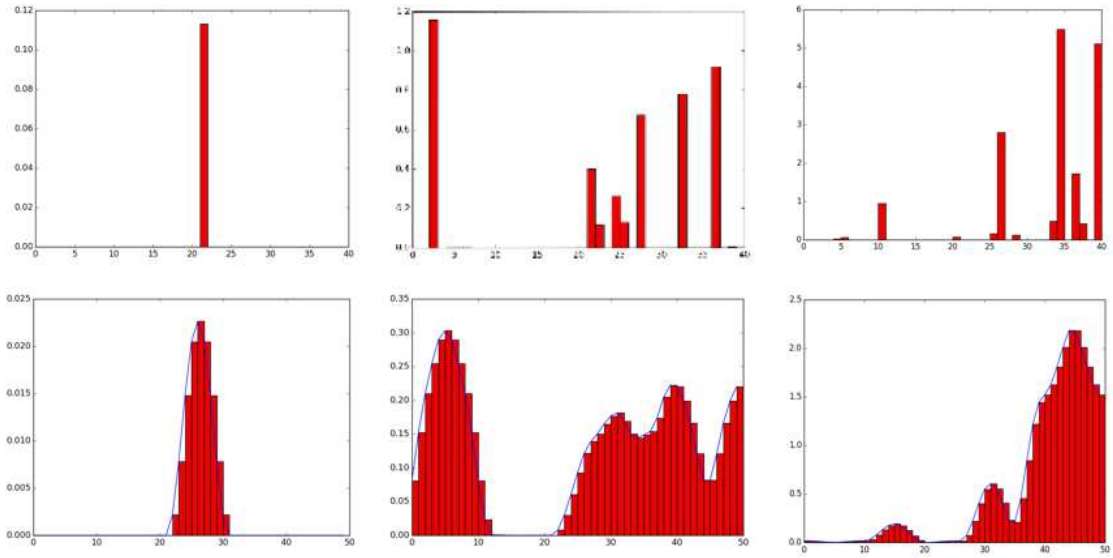


FIGURE 3.4: Sequence of flow fields can be noisy or discontinued. Smoothing time-varying flow fields signals enriches information about motion and transform discreet signal into near continuous signal.

sum of each packet E3.

3.1.2 Curved bell weighted cosine smoothing

Smoothing gives advantage such as giving robustness to outliers and improving generalization performance. Furthermore, Zero-th layer or depth of packet gram is smoothed in order to form continuous motion frequencies. It turns deeper layers Consider decomposition packet vector, we use Hanning window function f ((**??**)) which convert vector into taper formed using weighted cosine of such :

$$f(n) = 0.5 - 0.5cos(\frac{2\pi n}{M-1}) \tag{3.3}$$

where $0 \leq n \leq M-1$

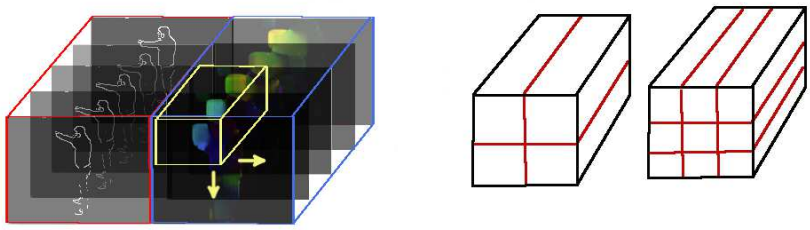


FIGURE 3.5: Two types of dense cuboid with spatio-temporal grid of either 2×2 and 3×3 regional division.

$$C_s = B_s \otimes f \tag{3.4}$$

In which is smoothed version following cosine bell curve shape of dynamic frequencies. Zero-th layer or depth of packet gram along sequence is convoluted ((**??**)) with f to produce final version of feature vector. As a compensation, dimension of feature vector is getting longer than original one. Mostly found on signal processing literature, it is used as windowing function for smoothing purpose. Wavelet Packet vector of no smoothed and smoothed version can be revealed in Figure **??**.

3.1.3 Temporal pooling as translation invariant

Wavelet is generally conforms translation and thus not translation invariant (Bruna and Mallat, 2013). The integral of coefficients of each sub band is deemed to be translation invariant. Thus, pooling of normalized feature vectors is one of method to be applied to each sub band in form of energy. Our

goal is to calculate the energy of each packet by making use of each coefficient. The energy of coefficients gives unique patterns for each scale or level and contribute to classification performance. If energies of wavelet packet in the certain level of every HOF bin are collected and concatenated with all regions within cuboid, it turns out to be final feature representation. We prove that such signatures bring generalization to describe motion cycles. Suppose wavelet packets for each level given m sub bands is w_m and number of the coefficient is c, the concatenation of sum of normalized w producing energy E in level l would be :

$$E_l = (\sum_{i=0}^{c} \|\mathbf{w_1}(\mathbf{i})\|, \sum_{i=0}^{c} \|\mathbf{w_2}(\mathbf{i})\|, ..., \sum_{i=0}^{c} \|\mathbf{w_m}(\mathbf{i})\|) \tag{3.5}$$

Final representation would be a complete set of all wavelet packet energies across the tree. Later, all of the concatenated energies for all bins in HOF will be integrated with all regions within cuboid. If we use 10-bin HOF, it means there are 10 channels, each of which collection of wavelet packet energies is measured. If we use 10 flow fields over time then it turns out to and we use cuboid with 9 regions, there will be 720-dimensional feature vector on the first level of decomposition.

3.1.4 Class specific Bag of Features and its cuboid sampling method

After HOFs are extracted, it is collected in sequence. Frequencies of each time frame within sequence along each bin of HOF can be treated as one-dimensional vectors. Sparse representation derived from Packet Flow energies within the spatiotemporal region by means dense cuboid. In order to adapt scale-invariant features, most researchers use multiresolution pyramid that forms scale factor based on the size of windows channel (Wang et al., 2011)(Laptev et al., 2008). More scaling factor enriches scale information

and boosts the performance but it is not significant and also there is increased computational cost that must be taken into account. Thus, we use $N \times N$ spatiotemporal cuboid divided into $r \times r$ regions to sample optical flow. Packet flows is obtained inside region of cuboid volume without any information about positions spatially and temporally but information of structure still be achieved, to this end, it is advantageous to adopt such region division cuboid. To enrich local geometrical information, the bounding volume is divided into four and nine dyadic regions as in Figure **??**. By concatenating all of each packet flows from regions inside cuboid, we gain bin of HOF, region and global level features. Every cuboid final feature vector U for r regions, cuboid and packet flow depth l will be inferred as:

$$U = (E_{l1}, E_{l2}, ..., E_{lr}) \tag{3.6}$$

where U is concatenated feature of the cuboid with r regions in total. Given final local geometric feature.

Dense cuboid patches are sampled through spatiotemporal video with predefined step size that is how many pixels cuboid patches slide. The high number of step size will burden computation, thus it is better to consider the size of $W \times W$ pixels step in spatial space (Wang et al., 2011) and T frames step along the temporal sequence (number of flow fields). While Packet Flow is multiresolution analysis inside regions of the cuboid, it is considered to choose large cuboid with high step size. The size of resulted feature vector dimension is independent of the size of window patches. Different size of the video (dimension) would influence the size of windows channels that reliable to cover action motion. The output of each cuboid patches is a Packet Flow feature vector. It would give same dimensional for all Packet Flows along the sequence of flow fields. In this research, cuboid is divided into 2×2 and 3×3 regions (r=2,3). Each type of region-based cuboid and every

depth of Packet Flow have its own dictionary in which learned separately to convert original L1-normalized Packet Flows features into k dimensionality of sparse representation by mean class specific sparse coding. This sparsity will reduce the influence of noise thus improve robustness against noise such as camera motions. By using matching and counting in such voting fashion, the sparse feature vector, later, is L1-normalized such that it will give rise on compact and fair deviation feature because every video appears in a different number of the sequence. Overall sequence of video, voting of codebook are sum pooled and then L1-normalized giving a final feature vector to be classified.

BoF has gained its popularity in computer vision topics. In this research, K-means clustering is used a dictionary generation. All the extracted PF features are matched, voted, and L1-normalized with a generated dictionary to form sparse features. Dictionary learning is constructed from training data and form D number class specific dictionary. Codebook $\{C_1, C_2, ..., C_n\}$ is built from a collection of Packet flows. Every class has its own codebook and cluster centers, for example class 0 has $\{C_{01}, C_{02}, ..., C_{0n}\}$. These sets of codebook vary depending on resolution attached to them in which accommodate multiresolution. This will give rise of scale invariant to the recognition system. Every feature vector generated from patches of the spatiotemporal flow field is matched with the generated codebook. The numbers of matching feature are quantized giving voting fashion. After such quantization, the feature vector is L1-normalized to give fair representation scale. The most prominent votes determine which parts of a region inside sequence of flow field space that has significant contribution to the motion. This will turn features into the aforementioned sparse representation that preserving local geometric information of features. These features are employed as feature input of classification algorithm.

FIGURE 3.6: Samples of KTH actions dataset which contains 6 classes

FIGURE 3.7: Samples of UCF Sports dataset with its 10 classes

3.1.5 Classification

We use SVM with generalized histogram intersection kernel (Boughorbel, Tarel, and Boujemaa, 2005) ((**??**)) for classification which is advantageous for BoF based features. Descriptors from histogram of BoF is normalized and trained by:

$$K(x, x') = \sum_{i=1}^{m} min\{|x_i|^c, |x_i'|^b\} \tag{3.7}$$

where

$$(x, x') \in X \times Y$$

$min\{|x_i|^b, |x_i'|^b\}$ is This kernel has positive definite as long as $b \geq 0$. We follow (Boughorbel, Tarel, and Boujemaa, 2005) to set b to 0.25 to be used along experiments. Because the nature of binary classification of SVM, classification framework is covered with one against the rest format and counting the highest score for final decision.

FIGURE 3.8: Samples of Hollywood 2 dataset which has 12 classes (8 classes is just on figure)

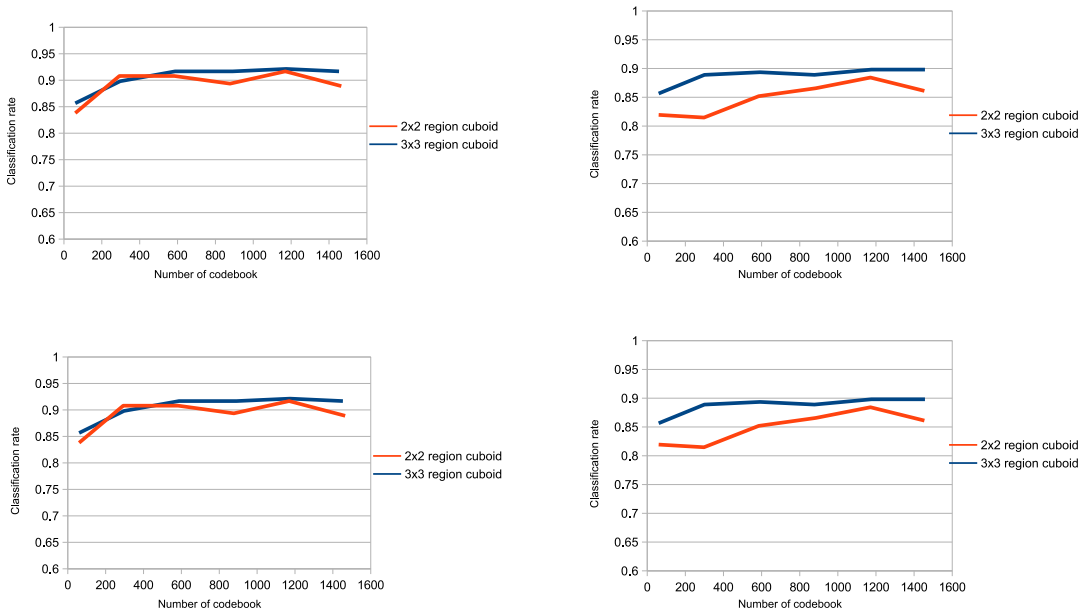


FIGURE 3.9: Effect of various codebook size on accuracy using packet flow of (a) fourth (b) third (c) second (d) first depth between 2x2 and 3x3 dense cuboid of KTH action dataset

3.2 Experiment

Dataset and parameter selection are two important set up for experiment purpose. The detail of dataset and selected parameter are described in detail in this section. Explanation on experiment setup will clarify and confirm results provided by proposed method. For cuboid sampling, we use size of cuboid of N=75 with spatial step size of W = 25 and temporal step size of T =10 which is less dense than (Wang et al., 2011) and less computation burden. Sampling steps is not totally overlapping and done for entire video samples. 15 flow fields per subsequence and 40 flow fields per subsequence

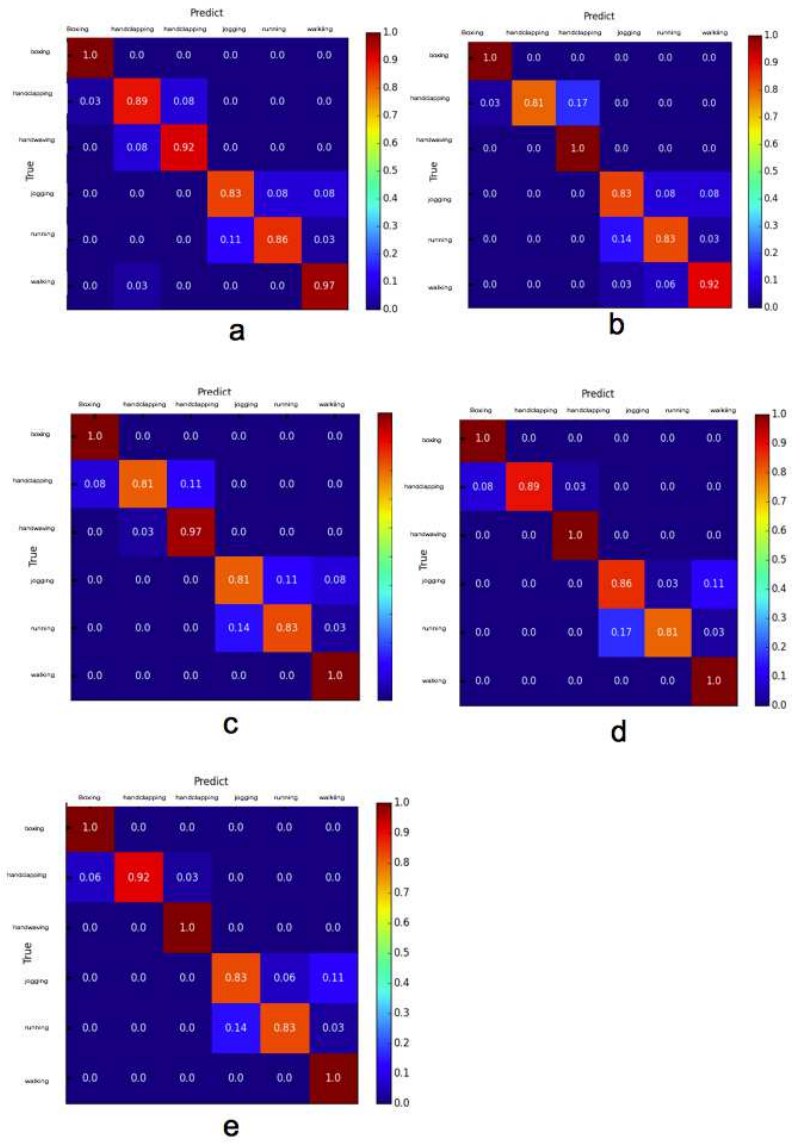


FIGURE 3.10: Confusion matrix of KTH classification results for packet flow of (a) first (b) second (c) third (d) fourth and (e) first and fourth concatenation depth.

of cuboid length are used for UCF Sports and KTH Actions respectively in consideration of each average video length.

3.3 Datasets

Dataset we use for entire experiment are KTH Actions dataset (Schuldt, Laptev, and Caputo, 2004) , and UCF Sports dataset (Soomro and Zamir, 2014). These are selected in consideration of size varying and conditioned or unconditioned dataset. The proposed method is tested to analyze its performance to recognize under different scales, color, illumination, occlusion, noise, background or foreground clutter, temporal dynamics etc. Performance evaluation for classification is done for the complete sequence of video.

There are 599 videos containing six action classes which are boxing, handclapping, handwaving, running, jogging, and walking (Figure **??**). It contains 2391 sequence in total. All videos have same dimension size of 160 x 120 and captured with 25 fps rate. 25 actors performed six different actions under different scales, illuminations, and viewpoints. For evaluation, there is much previous research that follows paper of origin setting by split dataset into train-validation (16 actors) and test (9 actors) or leave one person out validation (LOOCV) with different reasons. For the sake of fair comparison, train-validation-test split model which is same setting with the original dataset is used as evaluation for this paper.

UCF sports dataset is unconditioned action dataset gathered from sports events of various TV stations. There are 150 videos with a resolution of 720 x 480. It consists of Diving (14 videos), Golf Swing (18 videos), Kicking (20 videos), Lifting (6 videos), Riding Horse (12 videos), Running (13 videos), SkateBoarding (12 videos), Swing-Bench (20 videos), Swing-Side (13 videos), Walking (22 videos) as in Figure **??**. The speed of videos is 10 frame per second (fps) in average. For this dataset, we use evaluation setting introduced

TABLE 3.1: Classification rate of various bag of features based packet flow of KTH actions

Wavelet Packet depth	Number of features	Classification rate
Packet flow 1^{st} depth	2046	0.9120
Packet flow 2^{nd} depth	1762	0.8935
Packet flow 3^{rd} depth	2347	0.9000
Packet flow 4^{th} depth	2343	0.9256
Packet flow 1^{st} & 4^{th} depths	3808	0.9306

by **lan** in which uses specified train test split. The reason is by using LOOCV, there is a strong correlation between training and testing that make confuse whether features are derived from action or background. It is proved by increasing SVM parameter C (cost), accuracy is also increasing. For accuracy parameter, we use mean per class accuracy as a metric which is the same as baseline method (Lan, Wang, and Mori, 2011). It is counted by a number of true positives per total samples of each class per number of classes.

Hollywood 2 dataset contains 12 classes of human actions which are answering phone, driving a car, eating, fighting person, get out of the car, handshaking, hug, person, kissing, running, sitting down, sitting up and standing up. It is gathered by means of combination of script to video alignment and text-based script classification. It consists of 823 training video sequences and 884 testing video sequences. This dataset is challenging because it has complex background and various illumination degrees. Evaluation metric used in this dataset is mean average precision (mAP) of all classes.

3.4 Results

From Figure **??** we show correlation of codebook size and accuracies given either 3x3 region of cuboid and 2x2 region of cuboid of KTH actions dataset. We can measure whether Packet Flows is sparse enough or too sparse by investigating the graphics change. Even though distribution of both 2x2 and

TABLE 3.2: Comparison with another methods on KTH Actions

Methods	Accuracy
Global information (Wong and Cipolla, 2007)	86.6 %
Dense + HOF (Wang et al., 2011)	88.0 %
Cuboid + HOF (Wang et al., 2011)	88.2 %
Hessian + HOF (Wang et al., 2011)	88.6 %
Salient self similarity global features (Somasundaram et al., 2014)	89.6 %
Dense trajectories (Wang et al., 2011)	89.8 %
GRBM (Taylor et al., 2010)	90.0 %
3DCNN Ji et al., 2013	90.2 %
Dense cuboid + HOF (Wang et al., 2011)	90.5 %
Harris3D + HOG/HOF (Laptev et al., 2008)	91.8 %
Harris3D + HOF (Wang et al., 2011)	92.1 %
Deep learning SFA (Sun et al., 2014)	93.1 %
Our method (Packet flow)	93.1 %

3x3 based cuboid region are similar, they give different result on various codebook sizes. In general, 3x3 based gives slightly more accuracy compared to the 2x2 based because of more geometrical information available. This assumption does not hold in case, in contrast with KTH, there are many occlusion or background clutter which distract motions. For all diagram, almost 3x3 region based cuboid dominates accuracy rate for any codebook sizes. This could happen because there is abundance data sample for each class that helps build local geometrical instance. Moreover, there is little amount background clutter that possibly confuses between action motions and camera motions. It seems that sparser distribution gives better accuracy. The better accuracy result will be obtained if perhaps the size of dataset increases.

Classification rates of KTH actions dataset are shown on Table 1 using overall accuracy rate. Performance results given various Packet Flows are presented. It shows standalone (1^{st}, 2^{nd}, 3^{rd},4^{th}) and redundant (1^{st} & 4^{th}) Packet Flows results using concatenation of 2x2 based region and 3x3 based cuboid region each. Concatenation of 1^{st} and 4^{th} increases accuracy rate. Even though redundancy and accuracy difference is not significant compared

TABLE 3.3: Classification rate of various bag of features based packet flow of UCF Sports

Wavelet Packet selection	Number of features	mean per class accuracy
1	2324	0.7040%
2	948	0.6740%
3	2309	0.7030%
1,3	4633	0.6700%

with 1^{st} or 4^{th} alone, there is an improvement of classification rate. However, as more detail temporal dynamics are observed, it leaves uncertainty in which high degree of freedom arises. Later, it will be proved by comparing KTH dataset which has binomially 6 classes and high-density sample of each class and UCF Sports which has 10 classes but a low number of samples. Moreover, heterogeneous or homogeneous sample within class influences degree of freedom levels. The performance results compared to previous methods is presented in Table 2. Our proposed method is comparable to popular state of the arts. Most of state of the arts use flow-based features combined with another method.

The distribution of overall accuracies of KTH is shown in Figure **??** for every respective depth of Packet Flow. It shows confusion matrices that shows true positive rates for each class are obtained in different Packet Flow of which has 1^{st}, 2^{nd}, 3^{th}, 4^{th} and 1^{st} & 4^{th} depth respectively. The most confusion is between jogging and running even though it shows different distribution in depth. From these elaboration gives chance to select which temporal dynamic that is suitable to discriminate actions. From this result we infer that various resolution level it gives various performance.

For UCF Sports dataset, mean per class average is used as classification rate in consideration that different from KTH, UCF Sports classes are not well distributed, there are classes that have only a few members and another class with overload members. Besides, state of the art **lan** used mean per class average as classification accuracy of UCF Sports.

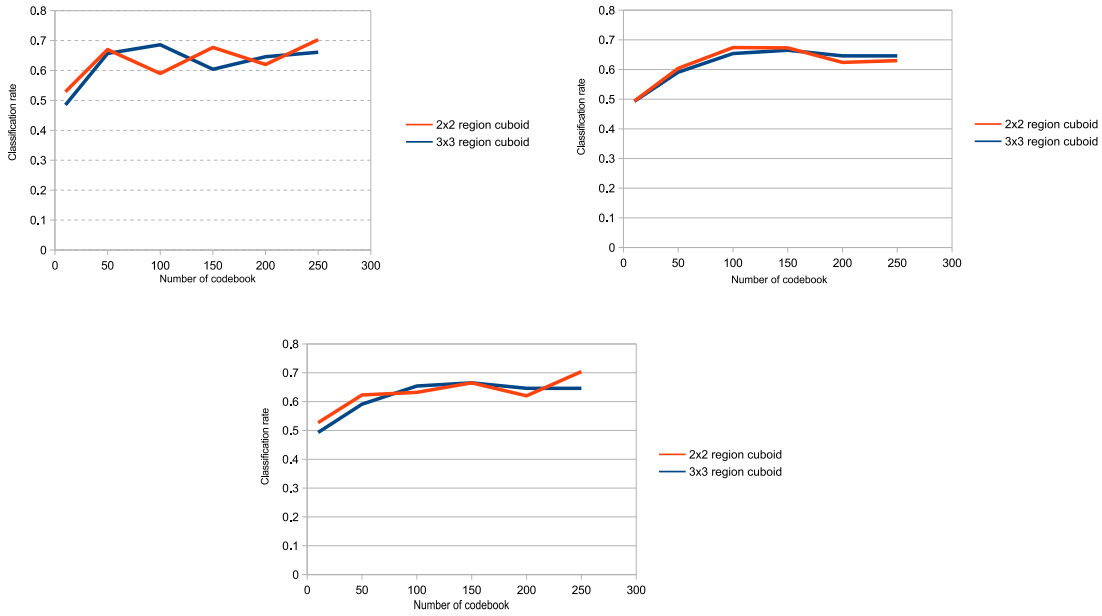


FIGURE 3.11: Effect of various codebook size on accuracy using packet flow of (a) first (b) second (c) third depth between 2x2 and 3x3 dense cuboid of UCF Sports dataset

By using sparse packet flow (BoF), we can identify flow in many degrees of depths. Every depth gives different characteristics such that we can select the depth that gives highest performance. There is a drawback that is the more Packet Flows are used there will be more potentially redundancy and exhaustive computation, however, It reveals good result for static or dynamic background and even for complex actions such as UCF Sports. Distributions of correlation between a number of codebook and classification (mean per class average) are presented in Figure **??** according to its Packet Flow's depth. Both 2x2 region and 3x3 region of cuboid compete each other along a number of codebook per class. The uncertainty arises probably because of high degree of freedom. Dynamic background caused by camera motions will reduce role of local geometrical information to discriminate action classes. However, we can choose the best cuboid which consists of Packet Flow's depth and region type of cuboid to be final representation.

Table **??** presents accuracy rate from each depth of Packet Flows. It shows different accuracy results obtained given 1^{st} depth, 2^{nd} depth, and 3^{rd} depth.

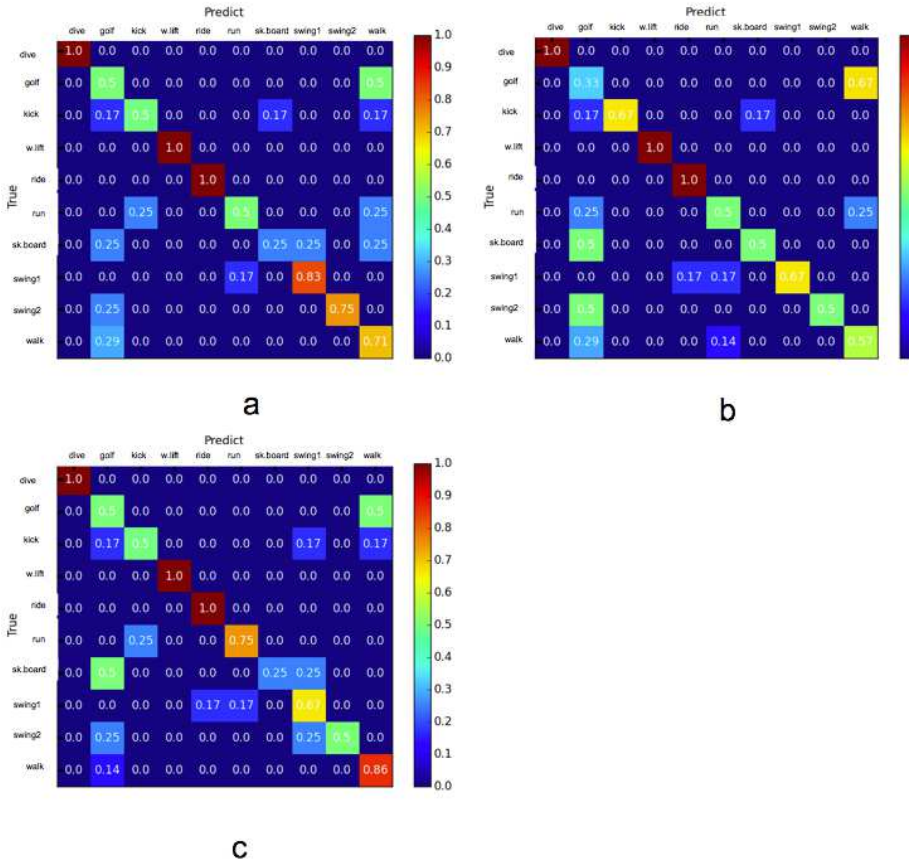


FIGURE 3.12: Confusion matrix of UCF Sports classification results for packet flow of (a) first (b) second (c) third depth.

The best accuracy rate is given by Packet Flows of 1^{st} depth with 0.704 accuracies. Combining the best of 2×2 region and 3×3 region cuboid does not help to boost accuracy result because of the probably high degree of freedom that turns distribution into uncertainty. If the size of the dataset is bigger, it means result will converge into certainty and help increase performance result.

Figure **??** shows confusion matrices of Packet Flows of 1^{st} depth, 2^{nd} depth, and 3^{rd} depth respectively. Diving, weightlifting, and riding give perfect classification result. That is because motion of actions is not biased with most camera movement and significant scale differences which confuse with actual action motions.

Compared to the state of the arts as in Table **??**, with either global and local BoF without localization, proposed method outperforms both. Even we

TABLE 3.4: Comparison with another methods on UCF Sports

Methods	Accuracy
global bag-of-words	63.1%
local bag-of-words	65.6%
spatial bag-of-words with $\Delta 0/1$	63.1%
spatial bag-of-words with $\Delta join$	68.1
Our method (Packet flow 3^{rd} depth)	70.3 %
Our method (Packet flow 1^{st} depth)?	70.4%

outperform with global and local BoF with classification loss ($\Delta 0/1$) and the joint loss of localization and classification ($\Delta joint$).

Extracting Hollywood 2 dataset requires extensive effort on extracting features because it can be sampled in various position in spatio temporal space. However using usual sampling of HOF, by using Packet Flow, it gains multi-resolution information in which in turn improve accuracy as in Table **??**. However, a number of used depths has to be selected by experiment to suitably choosing advantageous information. Based on experiment on Hollywood 2 dataset, it reveals that depth 1 and depth 2 has mutual information which gives best accuracy. In Table **??**, for comparison to the state of the arts, selected Packet Flows produces comparable mean accuracy precision compared to Deep learning SFA and another method.

UCF Sports is challenging dataset which is dynamic background potentially distracts motions. Moreover, actors appear in many scales and pose make recognition is difficult. Due to that, it is reasonable to localize action along spatiotemporal but since we do not consider localization we compare cuboid sampling with non-localized methods in which so far is only found in (Lan, Wang, and Mori, 2011) as in Table **??**. Compared to state of arts, our proposed method does not require space-time scaling and small either size of cuboid and step size sampling in which computationally expensive. Multi-resolution analysis that is nature characteristic of Packet Flow can spatially pool dominant motion and temporally extract detail temporal dynamics. It

leads to computational efficiency and better performance for action recognition. If localization is adopted and leave one person out cross-validation evaluation is used, higher accuracy is will be achieved.

Compared to state of the arts, Packet Flow can be extracted inside larger size of cuboid with larger step size. Table **??** shows comparison of cuboid sampling given feature points. It is less densely sampled than previous methods in which give comparable accuracies. The advantage is less computational effort while Wavelet Packet naturally has low complexity properties. Many researchers face trade-off difficulty between dense sampling and performance especially in action recognition where information sampling must be enough to be supplied to learning algorithm. It is proof that Packet Flow is spatially able to minimise non-dominant motion which is assumed to be noise and temporally to enrich temporal dynamics information. We found that without space-time scaling like (Wang et al., 2011) and (Laptev et al., 2008), comparable accuracy is still obtained by generalized SVM learning. We only give addition in cuboid division with either 2x2 and 3x3 region in which from each region Packet Flow is extracted.

3.5 Complexity

This method is implemented using Python and OpenCV. We use CPU of 3.7 GHz Quad-Core Intel Xeon E5, 12 GB 1866 MHz DDR3 ECC, and OSX platform for edge detection, dense optical flow, cuboid, HOF, and Packet Flow extraction phases. In our setting and our machine, our feature requires, overall, 1.3 frames per second to compute depending on a number of HOF bin, number sequence of HOF within cuboid or number of flow fields, step size of BoF, and how many Packet Flows to be counted. Based on entire experiments, HOF bin of 8, number of sequence of HOF within cubic of 40 for KTH and 15 for UCF Sports are used. As in Table **??**, compared to previous

methods, our proposed method is competitive since accuracy is high and complexity is low. We test on the video of which duration is 360 frames and resolution is 120 x 160.

TABLE 3.5: Comparison of cuboid sampling to state the arts

Methods	cuboid sizes	Sampling step size	cells	codebook	Accuracy
global bag-of-words with $\Delta 0/1$	$\{18\sigma, 18\sigma, 10\tau\}$ $\sigma = \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, 16, 16\sqrt{2}, 32$ $\tau = \sqrt{2}, 2$	50% overlap	9	1	63.1%
local bag-of-words with $\Delta join$	$\{18\sigma, 18\sigma, 10\tau\}$ $\sigma = \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, 16, 16\sqrt{2}, 32$ $\tau = \sqrt{2}, 2$	50% overlap	9	1	68.1%
Our method (Packet flow 3^{rd} depth)	$\{75, 75, 40\}$	$\{25, 25, 10\}$	4 and 9	2	70.3 %
Our method (Packet flow 1^{st} depth)	$\{75, 75, 40\}$	$\{25, 25, 10\}$	4 and 9	4	70.4 %

TABLE 3.6: Mean average precision (mAP) on Hollywood 2 dataset in respect to depth of Packet Flows

HOF	HOF + PF depth 1	HOF + PF depth 1 & 2	HOF + PF depth 1 & 2 & 3
0.46	0.475	0.512	0.482

TABLE 3.7: Comparison with another methods on Hollywood 2 dataset

Methods	Accuracy (mAP)
Dense trajectories (Wang et al., 2011)	47.7
Dense + HOG/HOF (Laptev et al., 2008)	47.7
Dense trajectories + HOF (Wang et al., 2011)	50.8
GBRM (Taylor et al., 2010)	46.6
Deep learning SFA (Sun et al., 2014)	48.1
Our method (Packet flow 1^{st} & 2^{nd})	51.2

TABLE 3.8: Complexity comparison with another methods on KTH Actions

Methods	Accuracy	Evaluation	fps
(Somasundaram et al., 2014)	0.8340	Train test split	0.6
(Fathi and Mori, 2008)	0.9000	Train test split	0.2-5
(Mikolajczyk and Uemura, 2011)	0.9530	LOOCV	0.12-0.18
Our method (Packet flow 1^{th} depth)	0.9120	Train test split	1.3
Our method (Packet flow 1^{st} & 4^{th})	0.9306	Train test split	0.5

Chapter 4

Motion Superpixels for Temporal Video Classification

Superpixels are a representation of still images as pixel grids because of their more meaningful information compared with atomic pixels. However, their usefulness for video classification has been given little attention. In this paper, rather than using spatial RGB values as low-level features, we use optical flows mapped into hue-saturation-value (HSV) space to capture rich motion features over time. We introduce motion superpixels, which are superpixels generated from flow fields. After mapping flow fields into HSV space, independent superpixels are formed by iteration of seeded regions. Every grid of a motion superpixel is tracked over time using nearest neighbors in the histogram of flow (HOF) for consecutive flow fields. To define the temporal representation, the evolution of three features within the superpixel region, namely the HOF, the center of superpixel mass, and the neighborhood correlation, are used as descriptors. The bag of features algorithm is used to quantify final features, and generalized histogram-kernel support vector machines are used as learning algorithms. We evaluate the proposed superpixel tracking on first-person videos and action sports videos.

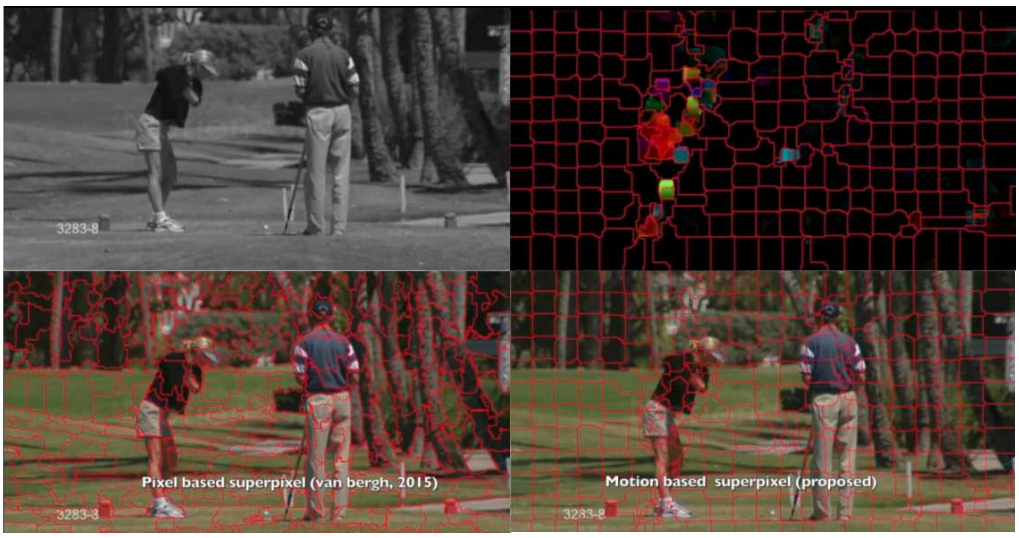


FIGURE 4.1: Top row: original image; motion superpixels on flow field. Bottom row: original superpixels from SEEDS; motion superpixels mapped to the original image (MPEG-4, 21 MB).

4.1 Motion superpixel

Motion superpixels are derived from motion space, in this case, from optical flows. The SEEDS algorithm creates segmented flows that are iterated using color distributions and boundary terms. Figure **??** shows the difference between spatial SEEDS, which oversegments RGB space, and motion SEEDS, which oversegments flow space. Motion superpixels are constructed where motion arises and remain in default form when there is no motion or very little motion, which depends on a threshold. If a motion superpixel is mapped to the original RGB space, then the superpixel will react if there is a moving object. We can filter out stationary superpixels by selecting superpixels for which the average from the HOF is greater than zero.

Motion superpixels can be any size and shape depending on the initial seeds in each frame. In Figure **??**, a dense optical flow is extracted from the original frame and mapped into HSV space. The flows from camera motion dominate the scene, which need to be compensated to obtain the actual motion triggered by an actor. Coarse to fine superpixel generation is important

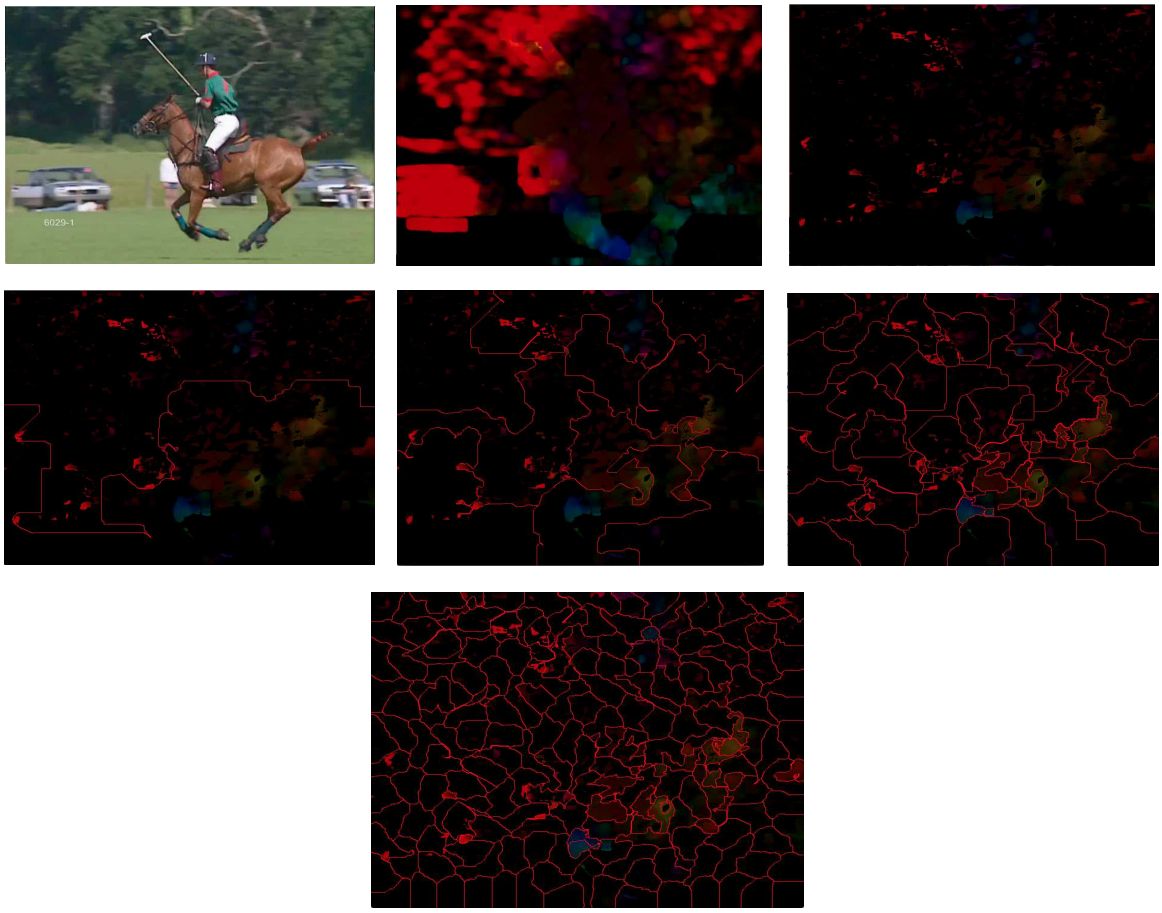


FIGURE 4.2: Top row: original frame of horse riding with camera motion following actor; dense optical flows; compensated motions. Middle row: motion superpixel from one seed; motion superpixels from 16 seeds; motion superpixels from 64 seeds. Bottom row: motion superpixels from 256 seeds.

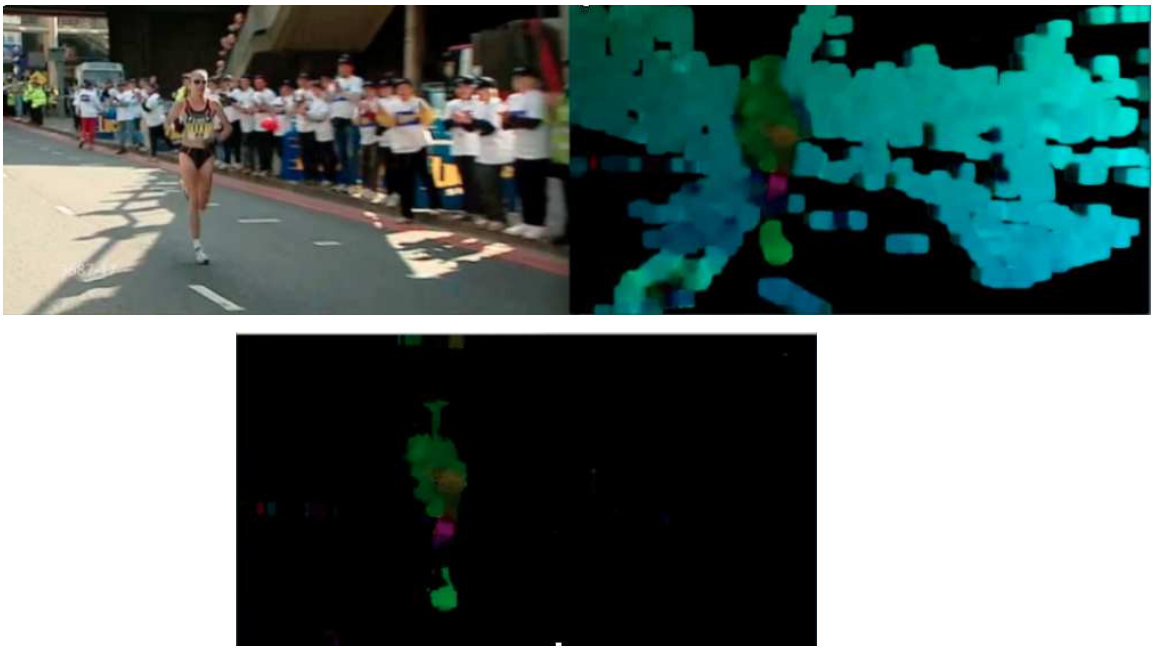


FIGURE 4.3: Clockwise from top left: original video of runner with camera following runner; flow field of optical flow which highlights the runner and the noise from camera motion; flow field after motion compensation to highlight only the runner?s actions. The motions of an actor or an object of interest are considered to be local motions.

not only to capture flows of varying size but also to add beneficial information for classification performance. This is especially true for videos that contain heterogeneous motions such as camera motion, object motion, non-object motion, and occlusion. We found that concatenating separate bags for different numbers of seeds captures different yet complementary information. This scheme is evaluated in the evaluation section using the UCF Sports dataset, which contains intense camera motions and occlusions that distract from flow-based features.

4.2 Motion compensation

Motions or flows within a frame are considered to be motion features. It is important to apply motion compensation because of camera motion. Video

captured using handheld devices have a high degree of freedom and often includes camera motion that distracts from actual motions. Moreover, it may be difficult to differentiate between actions of interest and the background from action videos taken in the field. To solve this problem, it is possible to use affine transformation and random sampling consensus. We use a similar consensus approach with a rigid transformation that estimates the affine transformation and removes those parts from the flow field. We use the rigid affine estimation in equation 1, in which i is a point inside the pixel region of the current frame X_t and the next frame X_{t+1}, and find a 2×2 matrix A and a 2×1 vector b that minimize the value of r:

$$r = argmin \sum_i ||X_{t+1}[i] - AX_t[i]^T - b||^2. \tag{4.1}$$

We find a transformation matrix from the reference frame to the next frame that represents the rigid transformation. In real action recognition, small camera movements can greatly impact flow alteration. By assuming that camera motion is rigid, the affine flow field can be removed from the flow field. After A and b have been found, the rigid prediction of X_t can be determined. If we assume that the rigid prediction is $\hat{X}_{t+1}$ then the following holds:

$$X_{t+1} = X_{t+1} - \hat{X}_{t+1}. \tag{4.2}$$

Figure **??** shows an original image of a runner, where a moving camera is following the runner. Only the salient movements are required to be processed during image recognition. In human motion analysis, articulated motion is considered to be salient as it creates varied and dynamic flows that cannot be modeled using affine transformations.

4.3 Motion superpixel tracks

While dense trajectories track a reference point over time based on flows, we use superpixels of flows, which contain more meaningful information and track reference superpixels over time based on the nearest moving point. Similarly, just as dense trajectories are tracked over multiple scales, motion superpixels are tracked in varying sizes. Grids for the initial superpixels are seeded in every frame and are iterated until convergence. Every convergent superpixel is then tracked with its corresponding superpixels based on the value nearest to its center of mass. Based on experiments, it is found that using 250 superpixel seeds per flow field is enough to give baseline results. We used five different values for the number of superpixel seeds (1, 4, 16, 64 and 256 initial seeds). Temporal information is prominent in activity recognition tasks. To treat motion superpixels over time, we used the nearest neighbor to find the superpixel corresponding to a specific super-pixel for times t to $t+1$ between two consecutive flow fields. To build the corresponding network between superpixels, we defined the center of mass for a superpixel region as follows:

$$m_{ij} = \sum_{x,y} A(x,y) x^j y^i, \tag{4.3}$$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}. \tag{4.4}$$

Equation (**??**) shows the spatial moments m_{ij} up to the first order for a polygon or superpixel, where $\{i,j\} \in (\{1,0\},\{0,0\},\{0,1\},\{0,0\})$ and $\{x,y\}$ represents points on the border of the superpixel. Equation (**??**) gives the center of mass $(\bar{x},\bar{y})$ of a superpixel. Note that each superpixel contains information about a set of motions. Each superpixel at time $t+1$ may have

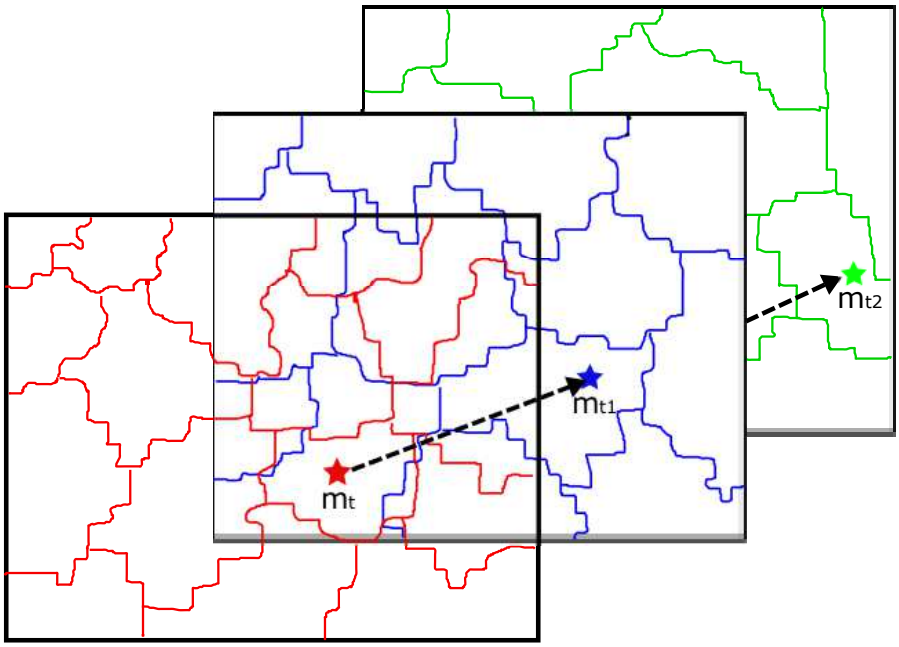


FIGURE 4.4: Tracking is done by following the nearest center of mass from a flow field frame m_t to the next frame m_{t1}.

moved based on motions contained inside the superpixels at time t. Therefore, the new central moments at time $t+1$ are the central moments at time t summed according to the average flow vectors $(\bar{v}_x, \bar{v}_y)$ in the superpixels at time t as described in equation 5:

$$m_s(t+1) = m_s(t) + (\bar{v}_x, \bar{v}_y). \tag{4.5}$$

Under the assumption that motion superpixels in consecutive flow fields will appear in a position nearest to its reference superpixel whether its shape changes or not, tracking is done by finding the nearest central moment point at each iteration. This is similar to tracking by following the path of the center of mass of superpixels over time as in Figure **??**. Along with several flow fields, this yields a time series of feature evolution. For each superpixel s, the next superpixel $s2$ is selected based on the minimum distance between the centers of mass:

$$d_{s,s2}(m_s(t+1), m_{s2}) = s \in S, s2 \in S2min||(m_{s2} - m_s(t+1)||. \quad (4.6)$$

After computing the optical flows, seeds of the superpixels are constructed and each superpixel is tracked based on its center of mass. A collection of centers of mass for the given time interval forms a sequence of tracked motion superpixels $(m_t, m_{t+1}, m_{t+2}, ...)$. The tracking is restricted to a time interval because longer track increase the probability of drift or bias from the initial point. In anticipation of this problem, we predefine the number of flow fields N in a sequence. If the next superpixel contains a flow field with all zero values or with no motion, then the sequence is terminated. If the length of the sequence is less than N, then the track is not saved as a feature. Conversely, if the track contains N flow fields and all superpixels contain motions, then the track is saved as a feature vector. In practice, a track length of $L = 10$ flow fields is used.

In general, superpixels with no motion are represented in HSV color space as zero values or in black. This means that there is no presence of motion in that superpixel region, or that the motion is removed because of motion compensation. We use the termination criterion of the absence of motion to prevent non-motion selection. As with dense trajectories (Wang et al., 2011), dense optical flows are more robust than sparse flows, and thus dense Farneback optical flows (Farnebäck, 2003) are chosen as the base flows for the entire process.

Local motion segmentation to form superpixels contains information about motion flows. A sequence tracked over time will produce a sequence of motion flows that can be described as a rich motion pattern. Given a number of flow fields N, the sequence of moments over time $M = (\Delta m_t, \Delta m_{t+1}, ...\Delta m_N)$ has the displaced central moments $m_t = (m_{t+1} - m_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$.

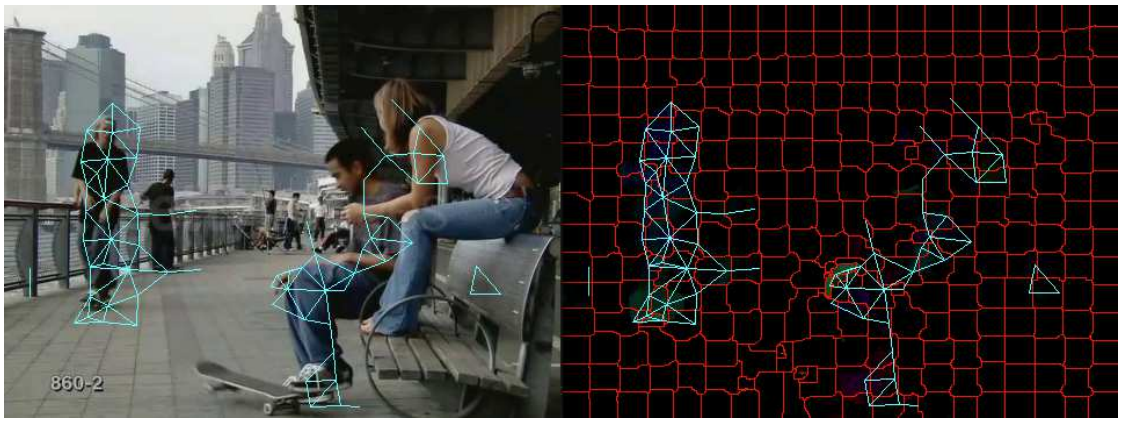


FIGURE 4.5: Edges of related motion superpixels that contain useful representations.

The resulting vector is normalized by the sum of the magnitudes of the displacement vectors in the central moments sequence:

$$S^* = \frac{(\Delta m_t, \Delta m_{t+1}, ...\Delta m_N)}{\sum_{i=t}^{t+N-1} ||\Delta m_i||}. \tag{4.7}$$

The vector S^* is the final normalized vector in the superpixel tracking representation. In this research, we only consider $N = 10$ because varying N does not significantly influence the accuracy of the results. However, this does not improve the results in practice. Thus, using a fixed number of flow fields produces a reliable final superpixel track vector.

4.4 Two-way motion superpixel tracks (bi-tracks)

Tracking along temporal flows requires path selection for dense superpixels. By using the nearest centers of mass over time, there is a possibility that a track will lose the most salient path for describing temporal evolution. This situation often happens when there are many centers of masses within a small distance in the next flow field. Selecting the nearest center of mass is not enough in this case, so we consider multiple selections to ensure that the generated paths are adequate for temporal representation. We select the

two nearest centers of mass at the first iteration. The first and second centers of mass are selected from the first and second flow fields, and only the nearest center of mass is selected in subsequent iterations. This produces two paths, which we call a bi-track, for every superpixel in the flow field. It can be helpful in the case of motion compensation, as in the UCF Sports dataset, to not remove camera motion, as this can confuse track generation. There is a possibility of adding more paths by using multiple selections of centers of mass to enrich feature sampling.

4.5 Feature descriptors

We define four features for evolutionary representations, namely the HOF, position (center of mass), the correlation with HOF neighborhoods, and the histogram of gradient (HOG). Every motion superpixel must form a region which consists of a boundary along with the flow field contained inside it. The HOF extracted from a flow field consists of several bins with direction and magnitude quantifiers. Superpixels are connected by edges formed by connecting centers of mass as in Figure **??**. It is possible to define a more holistic communal representation that changes over time. The communal representation of a set of neighboring superpixels is defined as the sum of correlations of the HOFs between one superpixel and its neighboring superpixels. For a superpixel s_e with neighboring superpixels $s_{2e} \in S_2$ depending on the edges $e \in E$, the local correlation is defined as

$$corr(s_e, S_2, i) = \sum_{e \in E} s_e[i] S_2(e)[i - j], \tag{4.8}$$

where i indexes the feature vector elements and j is the index difference. In this case, we set to zero only element pairs with the same index. We use

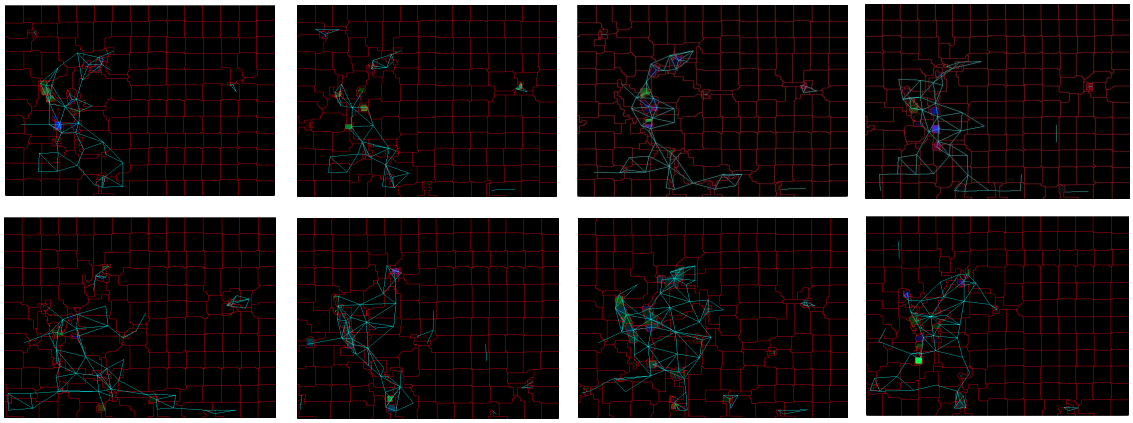


FIGURE 4.6: Temporal evolution of motion superpixels in sequence starting from the upper left corner to the upper right corner and from the lower left corner to the lower right corner.

HOFs with ten bins so that i takes ten values. By concatenating the correlations with respective to the index, this produces ten correlated elements, which is similar to correlation by convolution.

As in Figure **??**, the evolution of a superpixel's neighborhood can be captured by the evolution of correlations in the relevant region. The communal representation of neighbors changes over time, which gives dynamic information about moving objects or parts of objects.

4.5.1 Local Bag of Features & Classification

The BOF is constructed from data gathered by temporal sampling. The sampling step for ten flow fields is used as the subsequent feature vector. We use a class-specific dictionary formed from three iterations of k-means clustering. Fast k-means clustering (Bachem et al., 2016) is used, which is provably computationally cheap even without using a GPU. The number of clusters in the representation of features is usually determined heuristically by trial-and-error within a given range for the number of clusters. If the number of clusters is too small, then the representation may be shallow, whereas if the number of clusters is too large then the representation becomes nearly

flat and is hard to generalize. We consider using separate bags for different descriptors and different superpixel sizes. Previous research has shown that separating bags is advantageous, especially for different coding scales (Khan, Weijer, and Vanrell, 2012).

For classification, we use an SVM with a generalized histogram kernel, which has been shown to be robust in terms of quantification-based features such as histograms (Boughorbel, Tarel, and Boujemaa, 2005). The definition of a generalized kernel histogram is

$$K(x, x') = \sum_{i=1}^{m} min\{|x_i|^c, |x'_i|^b\}, \tag{4.9}$$

where x_i and x'_i are two different histograms that each contain m bins. This comparison is done for each element i and is iterated until there are m bins. The values c and b give generalized versions of the histogram kernel to handle fields of different sizes in histogram extraction. For example, the compared histograms are in general extracted from different sized superpixels. Based on Boughorbel et al. (Boughorbel, Tarel, and Boujemaa, 2005), we let $c = b = 0.25$, which gives good results in a large variety of contexts.

4.6 Experiments

Experiments are performed using the UCF Sports dataset (Rodriguez, Ahmed, and Shah, 2008), (Soomro and Zamir, 2014) in which the scenes are shot under real conditions for sport events and include camera motion. The camera follows the actors? motion as the objects of interest. Actors also appear at various scales and their motion is freely articulated with occlusions, making this dataset challenging for action recognition. The frame rate in the UCF Sports dataset averages ten frames per second. The dataset contains ten action classes (diving, golf swing, kicking, lifting, riding a horse, running,

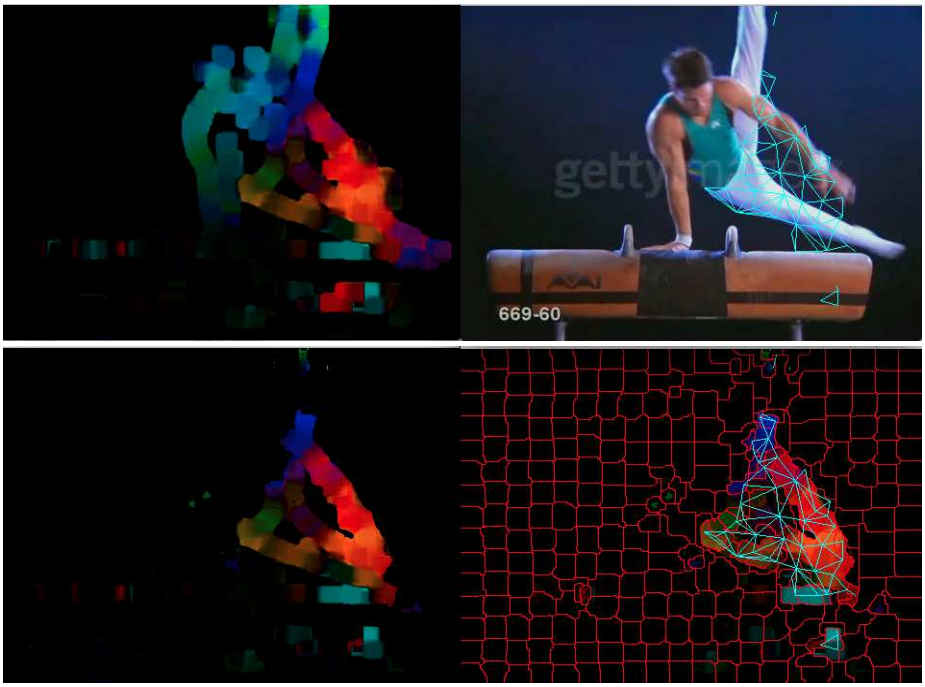


FIGURE 4.7: Motion superpixels for gymnastic movement.

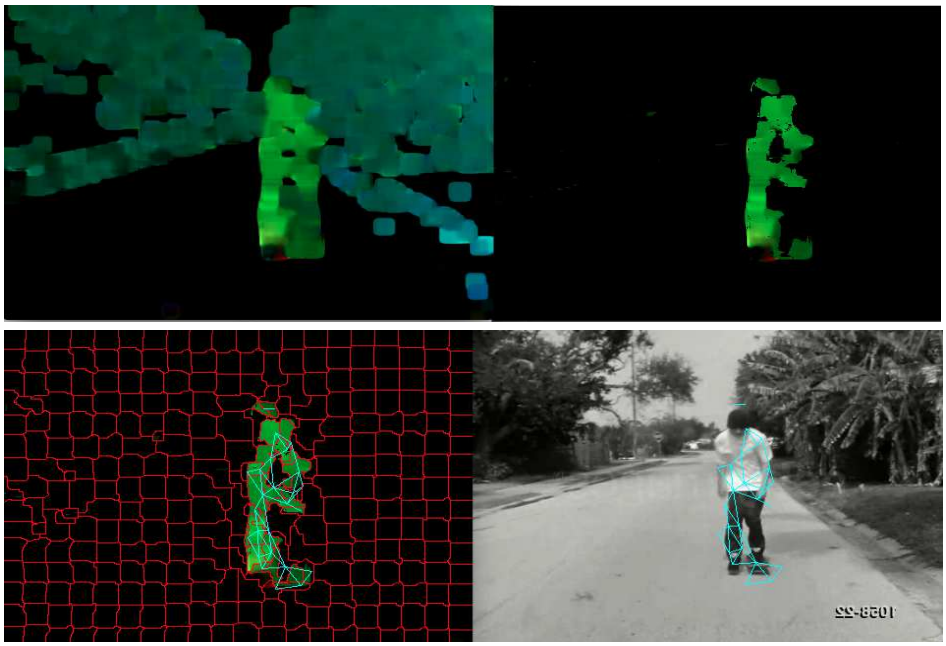


FIGURE 4.8: Motion superpixels for skateboarding.

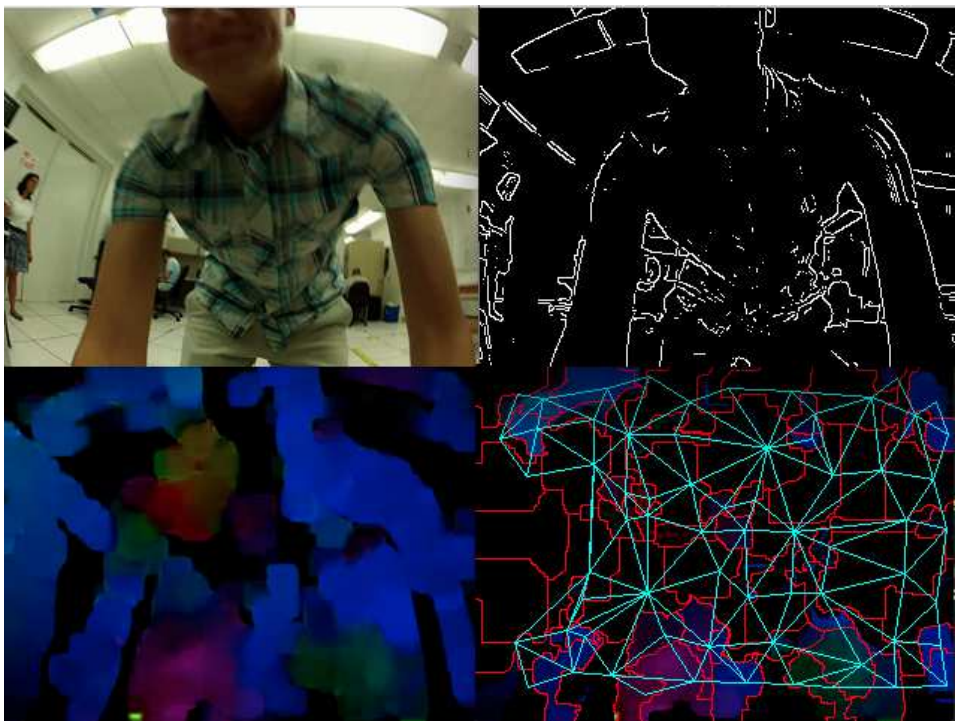


FIGURE 4.9: Motion superpixels for first-person video.

skateboarding, swing-bench, swing-side angle, and walking) with a resolution of 720×480 pixels.

We also try to evaluate the JPL First-Person Interaction dataset **ryoo** which is challenging because of the robotic vision. Our framework is suitable for adaptation to active vision, because the vision of a robot changes dynamically and has characteristics that are similar to camera motion. There are seven activity classes to be differentiated, and there are positive, neutral, and negative interactions. In particular, the classes are shaking hands, hugging, petting, waving, pointing, punching, and throwing. Figures **??** and **??** show example of features generated from the UCF Sports dataset, and Figure **??** is generated from the JPL Interaction dataset.

For feeding features into the BOF algorithm, we sample five flow fields of length ten for both the JPL Interaction dataset and the UCF Sports dataset. Because of considerations of computing time, the number of initial seeds for the UCF Sports dataset is 16, 32, 64, and 128, while for the JPL Interaction

dataset we use 50,150,250, and 350 and 1, 4, 16, 64 and 256. We found these sizes to be adequate for giving reasonable results. In specific cases, we try to explore how coarse or fine the flow information should be and how much locality of flow information is important for recognition in the JPL Interaction dataset. We also evaluate the datasets using tracks of length $L = 10$ and $L = 20$ to examine differences in the recognition rate. To this end, varying and fixed sizes of superpixels are compared to demonstrate the influence of large and small superpixels. The number of iterations of SEEDS is set into 50. A greater number of iterations will construct a more precise superpixel, but more computational time is required. Several experiments show that the gain in precision is not significant. We also tried various numbers of initial superpixels, but again the accuracy does not change significantly. For evaluation, leave-one-person-out classification is used for the JPL Interaction dataset, meanwhile leave-one-sample-out classification is used for the UCF Sports dataset.

For the experimental setup for the JPL Interaction dataset, we first decide how many clusters to compare between compensated and uncompensated motion. It is found that 1400-2800 clusters are suitable to give reliable accuracy with 150 and 250 seeds, as in Table **??**. With 1400 clusters there are 200 clusters for each class (seven classes), while with 2100 clusters there are 300 clusters for each class. Because the number of extracted features is 100000 on average, a reliable accuracy is achieved when the number of clusters per class is around 350. Therefore, we use the square root of the number of extracted features as the total number of clusters for the JPL Interaction dataset. For the UCF Sports dataset, a quarter of the square root of the number of extracted features for each class is used for the number of clusters, and therefore the total number of clusters in the codebook is the square root of the number of extracted features multiplied by the number of classes.

To confirm that motion compensation is important, we compare results

TABLE 4.1: Number of codewords relative to accuracy

Cluster number	700	1400	2100	2800	3500
Accuracy	0.78	0.82	0.82	0.82	0.78

TABLE 4.2: Effect of motion compensation

Superpixel seeds	Compensated motion	Uncompensated motion
250 seeds	0.82	0.75
150 & 250 seeds	0.88	0.79
50 & 150 & 250 seeds	0.85	0.75
50 & 150 & 250 & 350 seeds	0.88	0.81

between compensated and uncompensated motion. Motion compensation has a significant impact in helping motion superpixels identify desirable features. Table **??** shows the differences in accuracy between compensated and uncompensated motion using various numbers of initial seeds.

Figure **??** shows confusion matrix results for given classes in the JPL Interaction dataset using $L = 10$. We conclude that using various superpixel sizes obtains significant results because it covers coarse to fine motions, multi-scale motions, and more sample features.

TABLE 4.3: Comparison with other methods for the JPL Interaction dataset

Method	Accuracy
Global motion descriptor	72 %
Local motion descriptor	69%
Global & Local +χ^2 kernel	82 %
Local temporal motion superpixels	88 %

Compared with other state-of-the-art methods, temporal motion superpixels achieve comparable results for the JPL Interaction dataset. Table **??** shows that motion superpixels are better than global and local descriptors obtained from existing research **ryoo**

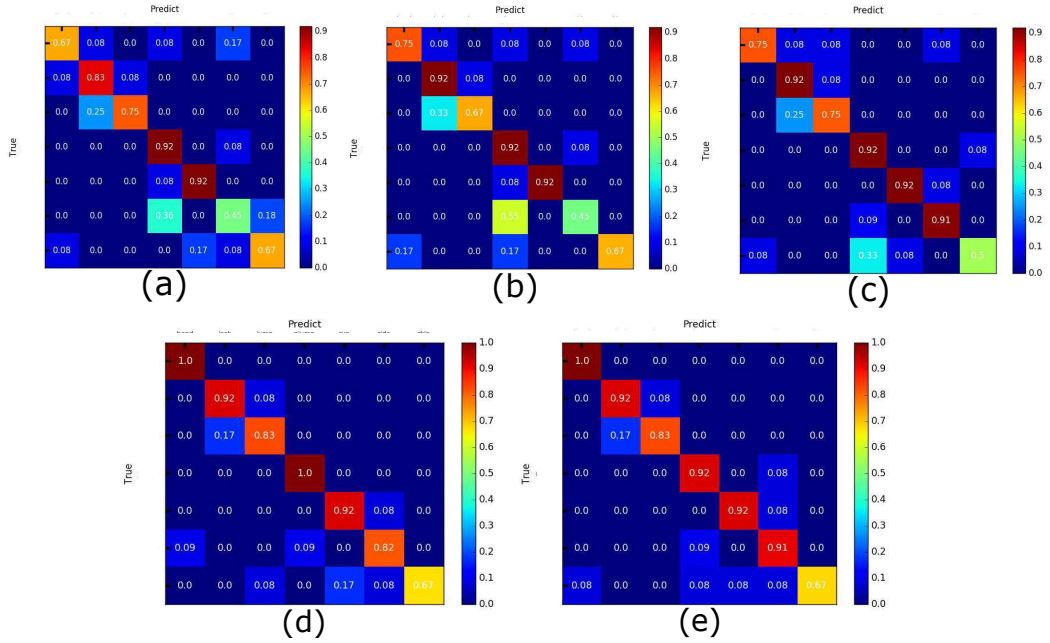


FIGURE 4.10: Confusion matrices for the JPL Interaction dataset using $L = 10$: (a) 1 seed; (b) 1 seed, 4 seeds; (c) 1 seed, 4 seeds, 16 seeds; (d) 1 seed, 4 seeds, 16 seeds, 64 seeds; (e) 1 seed, 4 seeds, 16 seeds, 64 seeds, 256 seeds. The mean accuracies are 67%, 79%, 81%, 84% and 86%, respectively. The vertical axes represent predictions, and the horizontal axes represent the truth; from top to bottom and left to right are shaking hands, hugging, petting, waving, pointing, punching, and throwing.

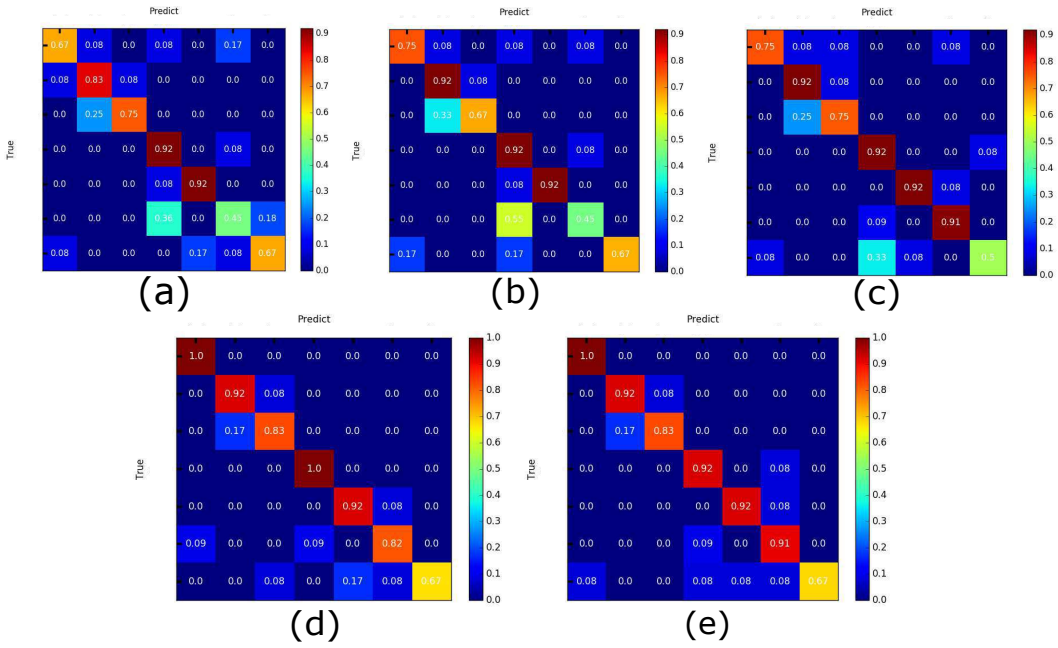
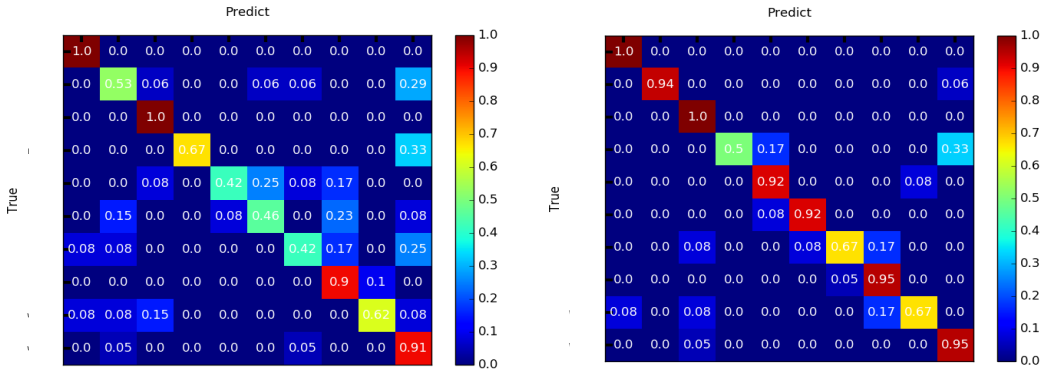


FIGURE 4.11: Confusion matrix for the JPL Interaction dataset using $L = 20$: (a) 1 seed; (b) 1 seed, 4 seeds; (c) 1 seed, 4 seeds, 16 seeds; (d) 1 seed, 4 seeds, 16 seeds, 64 seeds; (e) 1 seed, 4 seeds, 16 seeds, 64 seeds, 256 seeds. The mean accuracies are 74%, 76%, 81%, 88% and 89%, respectively. The vertical axes represent predictions, and the horizontal axes represent the truth; from top to bottom and left to right are shaking hands, hugging, petting, waving, pointing, punching, and throwing.



(a) HOF, correlation (b) HOF, correlation, HOG

FIGURE 4.12: Confusion matrix results for leave-one-out classification on the UCF Sports dataset. The vertical axes represent predictions, and the horizontal axes represent the truth; from top to bottom and left to right are diving, golf swing, kicking, lifting, riding a horse, running, skateboarding, swing-bench, swing-side angle, and walking.

Figure **??** shows confusion matrix results for motion superpixel tracks with $L = 20$. As in Figure **??**, more varied and detailed superpixels give better recognition rates, which indicates that locality is important for capturing flow representations. Comparing Figures **??** and **??** shows that $L = 20$ gives better confusion results for all superpixel sizes. This reveals that longer tracks increase the information gain of time series, although this increase is not significant.

Figure **??** (a) shows the confusion matrix results for the UCF Sports dataset. Compared with the JPL Interaction dataset, the UCF Sports dataset has a larger frame size of 720 x 480. Therefore, the initial number of seeds is set to 16 rather than one. The best confusion accuracy is 79% when using HOF and correlation descriptors and a single track. Figure **??** (b) shows that there is an improvement when spatial information is added (HOG) and bi-tracks (two-way paths) are used. This confirms that both motion and spatial information are important descriptors for activity recognition. Moreover, the use of bi-tracks enriches track information, which could be important for motion

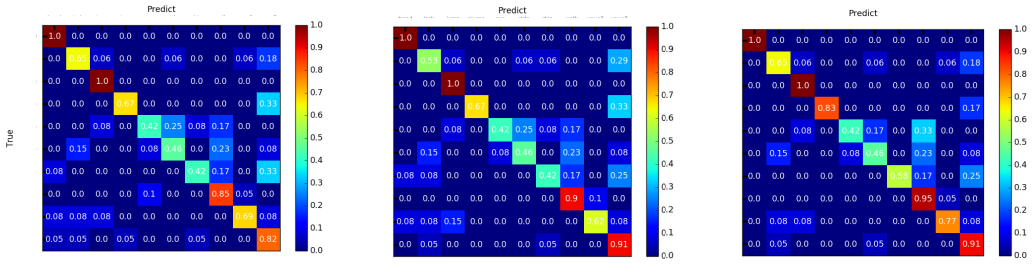
analysis.

TABLE 4.4: Comparison with other methods on the UCF Sports dataset.

Methods	Accuracy
STIP Sampling foreground only	71.92 %
STIP Sampling background only	73.97%
Dense sampling of STIP	75.34 %
Spatial superpixel (HOF, HOG)	86.7 %
Spatial superpixel (HOF)	87.9 %
Dense trajectories (HOF, HOG, MBH)	88.9 %
Temporal motion superpixel (HOF, correlation)	79 %
Temporal motion superpixel (HOF, correlation, HOG) bi-tracks	89.1 %

Table **??** shows a comparison of features sampled using non-geometric information such as cuboids. As opposed to superpixels, which give structural information about pixel boundaries, cuboid sampling is based on an arbitrary fixed size of the cuboid. We also compare with spatial superpixels (Dong, Tsoi, and Lo, 2014) for superpixels via RGB images without tracking or dense trajectories (Wang et al., 2011) for pixel-level tracking. We achieve the best results using spatial superpixel (using HOF descriptors) and dense trajectories (using HOF, HOG, and motion boundary histograms (MBH)). The difficulty with our approach is that even though superpixels contain more meaningful information, motion superpixel tracking paths have many possibilities and there is a high probability of losing the optimal path, thus requiring enrichment from the multi-track approach.

Figure **??** shows confusion matrix of LOO classification result of UCF Sports under various spatial wavelet packet decompotitions. The best average classification accuracy is 0.79 (fig. **??** c) with 2 layers spatial wavelet packet decompositions. It is shown that original features without decomposition achieves 0.74 (fig **??** a) while using 1 layer decompositions achieve 0.74 (fig. **??** b) accuracy which is same as original. Deeper decompositions are extracted, more scale invariant information is obtained. Running and sketboarding classes are most confusing class because there are many actors that



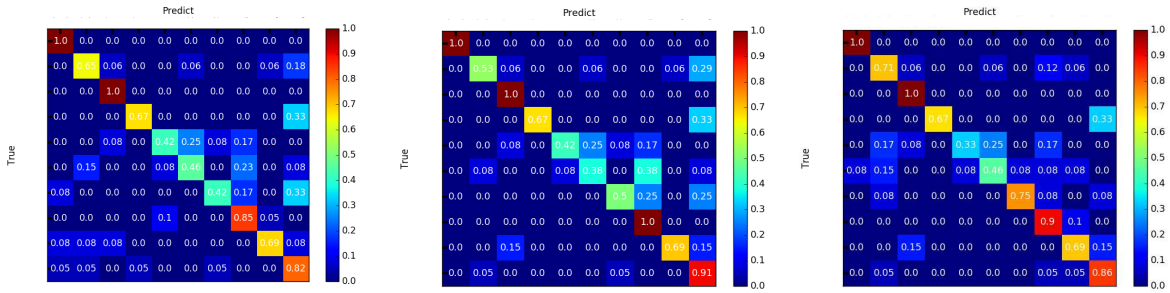
(a) original (b) 1 layer (c) 2 layers

FIGURE 4.13: Confusion matrices of classification result using leave one out of UCF Sports dataset of features with spatial wavelet packet decomposition except the original one. Vertical part is predict, horizontal part is true, from up to down or left to right are dive, golf-swing, kick, lift, riding, run, skate, swing-bench, swing-sideangle, and walk respectively.

are too small to be captured by initial number of dyadic superpixels. There is possibility to use finer-grain superpixel however computationally too expensive. Overall, UCF Sports dataset is heavily ill-conditioned with camera motion in which disadvantageous for motion based features such as optical flows.

Figure **??** shows confusion matrix of LOO classification result of UCF Sports dataset under various temporal wavelet packet decomposition. The best average classification accuracy is 0.77 (fig. **??** c) with 2 layers temporal wavelet packet decomposition. 10, 20, and 30 sequence of tracks are utilised to enrich temporal information. This setting is different from spatial one which only use 10 sequence of tracks. Fig **??** a and **??** b are original features and features with 1 layer wavelet decomposition respectively. It concludes that deeper temporal wavelet packet decompositions bring invariant to temporal dynamics by enriching various time interval into sampled features. In this case, even 10, 20, and 30 time interval tracks give 60 feature vector dimension.

Figure 8 shows confusion matrix of LOO classification result of JPL Interaction dataset under various number of superpixel seeds. It is shown that if



(a) original (b) 1 layer (c) 2 layers

FIGURE 4.14: Confusion matrices of classification result using leave one out of UCF Sports dataset of features with temporal wavelet packet decomposition. Vertical part is predict, horizontal part is true, from up to down or left to right are dive, golf-swing, kick, lift, riding, run, skate, swing-bench, swing-sideangle, and walk respectively.

variation of locality sizes is considered, higher accuracy is obtained. Various localities which are bounded by superpixels capture whole motions, part motions, or scale varying motions. Even though it is not significantly increase accuracy, it shows that there is no additional pattern, especially contribution of background (caused by camera motions), to help discrimination. Our local method outperform global motion and local motion (optical flows) using SVM with Gaussian kernel alone.

Chapter 5

Gated Spatio and Temporal Convolutional Neural Network for Activity Recognition: Towards Gated Multimodal Deep Learning

Activity recognition requires visual and temporal cues making it challenging to integrate these important clues. The usual schemes of integration are averaging and fixing the weights of both features for all samples. However, how much weight it needs for each sample and modality, is still an open question. A mixture of experts via gating Convolutional Neural Network (CNN) is one of the promising architectures to adaptively weigh every sample within a dataset. In this paper, rather than just averaging or fixed weights, we investigate how natural associative cortex like network integrates expert's networks to form of gating CNN scheme. Starting from Red Green Blue (RGB) and optical flows, we show that with proper treatment, gating CNN scheme actually works and sheds a light on information integration in future for activity recognition.

5.1 Gating CNN models

Very deep gating network is introduced to handle the noise and occlusion in scene for activity recognition. The proposed gating architecture can be adapted to different contexts depending on the purpose; i.e., gating network for handling integration of audio, text, image, object of various spatial resolutions, or actions with various temporal segments. This will enable the lower layers of the network to learn parameters with discriminative power. Furthermore, to the best of our knowledge, despite its simplicity, the proposed work is the first natural gating CNN to be introduced in video classification. We use gating network similar or shallower to the expert networks. For example, if the gating network is VGG-16, it means that both expert networks are also VGG-16, ResNet-50 or simple classifier for simplicity.

The use of deep neural network does not necessarily have to be specific model or size of CNN, however, recently VGG-16 and Residual Net (ResNet) are popular and achieve state of the art results on image classification (Simonyan and Zisserman, 2014a)(He et al., 2016). Thus, besides VGG-16, we use another popular network called ResNet-50. Figure **??** summarizes models consist of fusion by averaging, fusion by SVM and gating network. The reason of using various models is to compare possible fusion schemes including our gating CNN. Even though gating network model is similar to experts one, it is different in terms of output dimension. The dimension of gating output is 2, one is for weighing spatial expert and another is weighing motion expert.

In VGG-16 while making network deeper, convolution filter size is smaller which allow capturing coarse to fine pattern of images. For every output layer, non-linear activation function of rectified linear unit (ReLU) is used as it has shown better convergence properties and performance gains with little risk of overfitting. Another network models such as ResNet or Inception

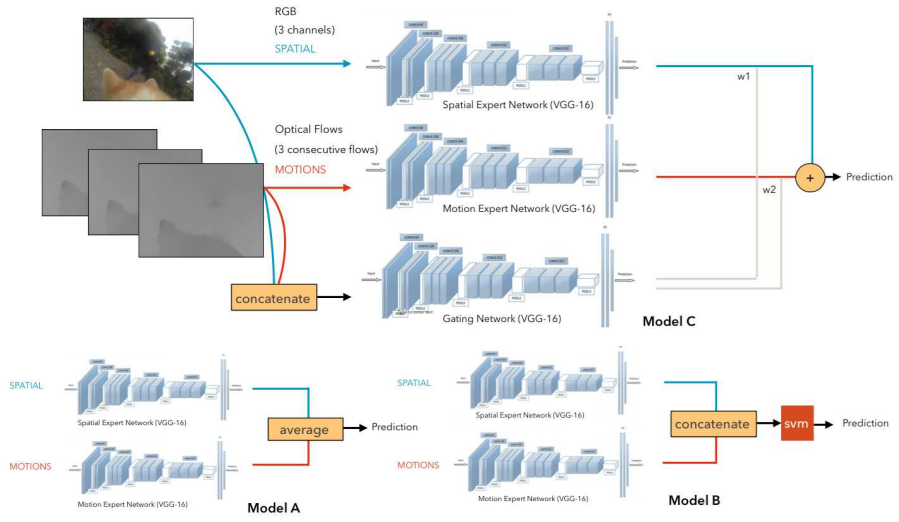


FIGURE 5.1: Various models of gating scheme.

(a) fusion by averaging (b) fusion by concatenation and SVM (c) gated network similar to expert network

is possible to be chosen and possibly achieve higher accuracy with saving memory. However, for training gating scheme, VGG-16 and resNet-50 are suitable as starting point.

5.2 Gated bi-modal CNN Design

We briefly introduce Gated CNN in Section **??** about the pipeline of gated bi-model CNN. Section **??** explains about general framework about gated CNN. Section **??** explains about training and testing scheme. Section **??** considers various combination of gating architecture.

5.2.1 Expert-Gating Pipeline

Training gating network can be applied in two-fold, by parallel learning both experts and gating network or sequential learning by training experts first and then gating network. For training gating network and expert network at the same time, it requires careful initial parameter setting. For example,

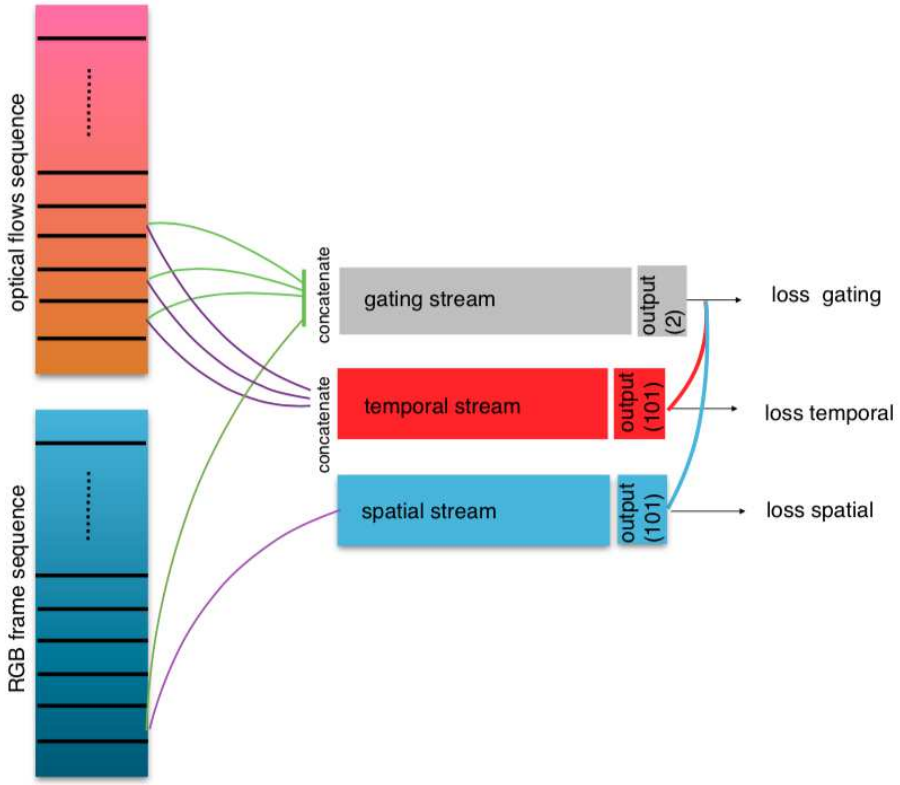


FIGURE 5.2: Expert-Gating training framework.

we have to ensure that during training, spatial expert network and motion expert network do not exceed each other in terms of accuracy, thus, gating network can learn true prediction data in sufficiently. Specifically, we use learning rate of 0.000001 to spatial stream because it tends to converge significantly faster compared to the motion stream. This is due to higher match between RGB frames with pre-trained data, Imagenet. For motion stream, we use learning rate of 0.0001 as those combination is sufficient enough to stabilize running in order the gating stream to be able to train enough data. However, to tune this type of learning for both expert streams and gating stream is trivial and the result is somewhat not optimum. For instance, if spatial expert network has reached 10% increase in accuracy compared to motion expert network, it means learning rate of spatial one must be slowed down in order to balance the gating scheme. Rather than this type of learning, we consider splitting the data to train expert network first followed by

gating network and continue learning after gating is trained. It can be summarized into this pipeline :

1. Random video frames are selected, thus for every iteration is given different input frame. RGB frames are inputted into spatial network while Flows are inputted into temporal network. The gating stream is inputted using concatenation of RGB and Flows for the sake of competitiveness between both modalities.

2. Given input modalities, each expert is trained independently until it converges.

3. Gating network is trained until the loss is stagnant.

4. On testing, gating output weighs each expert's output and fuse both weighted outputs. Then classification is done by selecting maximum value within dimension as predicted label

5.2.2 Framework Overview

The input of the gating network is concatenation of the spatial and motion information. Each stream has its own loss function that is updated independently, as shown in Figure **??**. The gating mechanisms such as the input gates and output gates follow this equation:

$$y_{final} = x_1 y_1 + x_2 y_2 \tag{5.1}$$

where:

$$y_1 + y_2 = 1 \tag{5.2}$$

where x_1, x_2, $y = (y_1, y_2)$, and y_{final} are the outputs of the RGB stream, optical flow stream, gating stream, and final prediction, respectively. This fusion scheme is presented in Figure **??** model C and in Figure **??** in detail. The

output gate is an additional fully connected layer with 101 inputs and two output dimensions. This structure is considered because, in nature, VGG output is 101-dimensional for UCf-101 and 51 for the HMDB-51 dataset (trained on ImageNet with 1,000 classes). The final fusion of the output of the expert streams is then normalized using a softmax cross entropy function. Furthermore, for the output of the gating stream, a softmax function is used to transform every feature vector's element as a float between 0 to 1 while the sum of a y_1 and y_2 is 1.

5.2.3 Input, Training-Testing Scheme, and Loss Function

Learning consists of two parts: expert learning and gating learning. To train the gating network, experts must be trained and produce feature vectors so that the gating CNN can estimate the proportion of each network relative to the other.

Dataset: We split the training dataset into half: the first half is for training the expert networks and the other half is for training the gating network. However, the whole dataset is used to train the expert network once the gating networks have been trained.

Input and Data Augmentation: The frame selection for each iteration is randomized. Hence, for every iteration, the method selects a different frame for the same video, thus training on all the frames as it iterates. Three networks are used for this gating CNN scheme; hence, there are three inputs: RGB for the spatial expert network, optical flow for the motion expert network, and a concatenation of RGB and optical flow for the gating network. In this case, RGB contains three channels and the optical flow contains three consecutive flow fields over time with two flow field differences. Therefore, for the gating network input, there are six channels for the first layer of convolution. The optical flow representation is basically transformed into

a gray-scale image; thus, three consecutive flows give the same amount of input as the RGB. To overcome overfitting, various pre-processing schemes such as cropping and flipping were performed. We used four-corner cropping and center cropping along with flipping. All the inputs were resized to a resolution of 250 $\times$ 250 with an arbitrary cropping of size 224 $\times$ 224 along with a horizontal flip. A mean image size of 250 $\times$ 250 was computed for the training set and used to subtract all the images.

Training the expert CNNs: For the spatial stream, pre-trained ImageNet was used to reduce overfitting. This kind of transfer learning has improved accuracy by a large margin. For the motion stream, network was trained from scratch because optical flow features are clear enough to define action, in contrast to spatial scene information. Whether the pre-trained ImageNet model or an untrained model is used initially, the effect on test accuracy and overfitting is still the same for the motion stream. For VGG-16 and ResNet-50, we used a learning rate of 0.001 for the spatial streams. It decreases to 9/10 of its value every 5,000 iterations with a momentum of 0.9. The maximum number of iteration was set as 20,000. For the temporal streams, we set a smaller initial learning rate (0.0001) in our experiments. It decreases to 9/10 of its value for every 20,000 iterations and uses momentum of 0.9. The maximum number of iterations was set as 100,000. We also consider transferring the weight of trained expert streams for VGG-16 using the good practice approach from (Wang et al., 2015) and used (Wang et al., 2016) for the temporal segment network to be gated with our trained VGG-16 gating network. Note that our trained VGG-16 uses the Caffe framework.

Training the gating CNN: For the gating network, we initialized network weights with pre-trained models from ImageNet. Next, we trained using a learning rate of 0.001, which decreases to 1/10 of its value every 20,000 iterations. Based on experience, if we set the learning rate to a large value (e.g., 0.1), the network tended to choose one of the expert streams, which is not

desirable. Training a very deep network such as VGG-16 is computationally heavy and it takes a long time to converge. Training a simple classifier uses a learning rate of 0.001, reducing by 10 % every 5,000 iterations. The maximum number of iterations was set as 100,000.

Testing the gating CNN: For given video sequence, we sampled 25 frames equally spaced and fed every frame to its respective stream (3-channel RGB to the spatial stream and three consecutive flow fields to the motion stream) and paired RGB and optical flow into the gating stream. Each of 25 softmax output pairs were then weighted and averaged to predict class.

Testing the two good-practice streams: For a given video sequence, we sampled 25 equally spaced frames and fed every frame to its respective stream and paired RGB and optical flow into the gating stream. The gating output weighted all 750 softmax cross entropy outputs and then averaged to predict the classes. For every frame in the spatial sequence, there were five crops (four corners and one center) with horizontal flips, thus 10 images were generated for every frame and $25 \times 10 = 250$ were generated for every sequence. Optical flow only formed the center of 10 stacks of three consecutive flow fields for 25 images in a sequence multiplied by two, thus generating $50 \times 10 = 500$ images in total.

Testing the temporal segment network: For given sequence of video, we sampled 25 equally spaced frames and fed every frame to its respective stream and paired RGB and optical flow into the gating stream. All 25 softmax outputs were then averaged to predict class. For every frame, for the spatial sequence, there were five crops (four corners and one center) with flipping; thus, the number of images for every frame was 10. Optical flow only formed the center of 10 stacks for 25 sequences, thus their total was $25 \times 10 = 250$.

Loss function: We used a separate loss function for the expert and gating networks. However, both have basically the same loss function, which

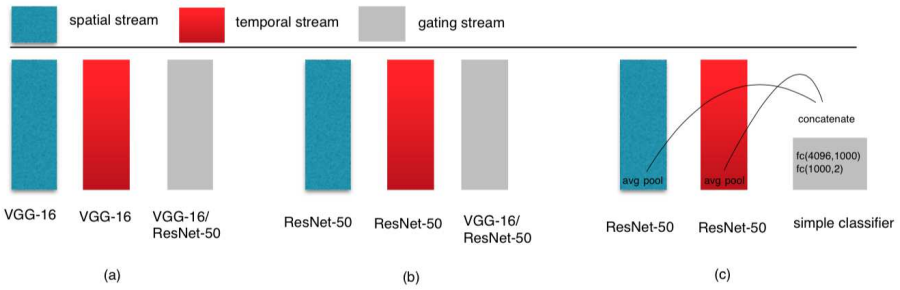


FIGURE 5.3: Various combinations of expert-gating CNNs.

(a) VGG-16s as the experts & VGG-16/ResNet-50 as the gating (b) ResNet-50s as the experts & VGG-16/ResNet-50 as the gating (c) ResNet-50s as the experts & a simple classifier as the gating

minimizes the error of the predefined labels. For the gating network, back-propagation tried to minimize the loss of the gated feature vector using the following softmax function:

$$E = -\sum_{i} t_i \log o_i \tag{5.3}$$

where o is the softmax cross entropy of output network v:

$$o = softmax(v) \tag{5.4}$$

The gradients with respect to the feature vectors at the last layer were computed from the contrastive loss function and backpropagated to the lower layers of the network. Once all the gradients were computed at all layers, we used minibatch stochastic gradient descent to update the parameters of the network.

5.2.4 Various expert-gating CNN combinations

The base of the expert network can be either two streams of VGG-16 or two streams of ResNet-50 with its gating. A gating CNN also has many

possibilities; however, to keep pace with the expert networks, the gating stream should have the same capability as the expert stream. The architecture of the gating itself is still an open question; however, a combination of deep and shallow networks (a simple classifier) can reveal its drawbacks and strengths. Therefore, we prepared several scenarios for expert-gating combinations. VGG-16 has 16 layers while ResNet-50 has 50 layers. We assume that deeper network will increase the number of degrees of freedom, which distracts the network from reaching the optimum solution. As shown in Figure **??** a, VGG-16 streams can be attached using a ResNet-50 or VGG-16. Figure **??** b shows that ResNet-50 streams are gated with ResNet-50 or VGG-16. Figure **??** c shows that ResNet-50 streams are weighted by a simple classifier with an input size of 4,096 (the concatenation output of ResNet-50's last layer without the fully connected layer from both experts). The simple classifier consists of two layers with 4,096 inputs and 1,000 outputs followed by a layer of 1,000 inputs and two outputs.

5.3 Results & Discussions

5.3.1 Datasets & Experiment details

Two challenging dataset are used as evaluation setup which are UCF-101 (Figure **??**) and HMDB-51. It has challenging problem because the size of dataset is small in case of deep learning. UCF-101 consists of 13K videos with 180 frames per video in average and 101 classes. HMDB-51 consists of 6.8K videos and 51 classes. For training gating network, we use UCF-101 dataset split 1 and use that trained model for entire experiment which suitably increase accuracy for all cases. The split of training and testing scheme is based on THUMOS13 challenge (Jiang et al., 2014). For entire experiment, we only use split 1 as analysis of our gating network. We use stochastic gradient

FIGURE 5.4: UCF 101 dataset containing 101 action classes with 9,537 training videos and 3,783 testing videos

descent as optimizer for both experts and gating network. Due to the limited memory, we use mini-batch size equal to 12 with momentum of 0.9. The learning rate is set to 0.0001 and 0.001 for RGB and flow network respectively. For the extraction of optical flow, we choose the TVL1 optical flow algorithm implemented in OpenCV with CUDA. The whole training time on UCF101 is around 2 hours for spatial network, 18 hours for temporal network, and 6 hours for gating network with GPU TITAN.

During training of VGG-16 experts, after 40 epoch, training is stopped and gating CNN is trained using another half of training dataset and evaluation. After that, training is continued until 80 epoch and evaluation is run. Next, both experts are trained using whole of training dataset until convergence. We also use initial parameter copied from two stream trained on good practice of (Wang et al., 2015) of which we call VGG-16 good practice and temporal segment network of (Wang et al., 2016) to be gated with previous

trained gating.

5.3.2 Results

Our gating experiment steadily outperforms fixed weight scheme. Table **??** shows test accuracy after 40 epochs training. Gating VGG-16 and gating classifier give the best accuracy along with gating classifier in this state at 71.8 %. Gating ResNet-50 does not find the best solution even when the loss starts to converge. Gating network is only trained on this epoch while expert networks training is resumed. Table **??** shows that gating classifier still outperforms fixed weights even after training 80 epochs. However, in Table **??**, after the expert networks converge, only gating VGG-16 exceeds fixed weights, while simple classifier one overfits. Meanwhile, ResNet-50 has high degree of freedom that gives obstacle for gating network approaching optimum solution. After expert networks converge, train achieves nearly 90 % for both spatial and temporal network while testing 72 % and 76 % which is overfitting. On that situation, gating network cannot be trained because training dataset has already nearly saturated yielding large margin between training and testing accuracy.

Table **??** shows result of ResNet-50 experts, gating VGG-16. gating classifier also outperforms fixed weights at 40 epochs. With 80 epochs training as in Table **??**, gating VGG-16 also gives best result in weighting. Gating classifier obtains 77.80 % which is also exceeding fixed weights. However, after the training is finished and the difference between training and testing accuracy margin becomes large of 99 % to 78.08 % for spatial stream and 90 % to 74.89 % for motion stream, shallower network (classifier network) overfits testing data as in Table **??**.

We also try gating of two streams CNN with its weight transferred from good practice of (Wang et al., 2015), our gating VGG-16 shows best accuracy

TABLE 5.1: VGG-16 40 epochs on UCF-101 (split 1)

RGB weight	Flow weight	Test Accuracy
1.0	0.0	46.34 %
0.0	1.0	66.80%
0.9	0.1	53.27 %
0.8	0.2	58.72 %
0.7	0.3	63.60 %
0.6	0.4	67.54 %
0.5	0.5	70.05 %
0.4	0.6	71.48 %
0.3	0.7	70.69 %
0.2	0.8	69.18 %
0.1	0.9	69.18 %
gating VGG-16		**71.82** %
gating ResNet-50		67.54 %
gating classifier		**71.82** %

TABLE 5.2: VGG-16 80 epochs on UCF-101 (split 1)

RGB weight	Flow weight	Test Accuracy
1.0	0.0	65.47 %
0.0	1.0	69.66 %
0.9	0.1	71.21 %
0.8	0.2	72.04 %
0.7	0.3	73.55%
0.6	0.4	74.23 %
0.5	0.5	76.34 %
0.4	0.6	77.01 %
0.3	0.7	76.22 %
0.2	0.8	73.45 %
0.1	0.9	72.32 %
gating VGG-16		75.5 %
gating ResNet-50		74 %
gating classifier		**76** %

TABLE 5.3: VGG-16 300 epochs (already overfit) on UCF-101 (split 1)

RGB weight	Flow weight	Test Accuracy
1.0	0.0	72.45 %
0.0	1.0	76.33%
0.9	0.1	79.21 %
0.8	0.2	80.23 %
0.7	0.3	81.54 %
0.6	0.4	82.77 %
0.5	0.5	82.81 %
0.4	0.6	83.5 %
0.3	0.7	82.74 %
0.2	0.8	81.22 %
0.1	0.9	79.61 %
gating VGG-16		**83.5** %
gating ResNet-50		81.24 %
gating classifier		82.10 %

while also approaching optimum solution if compared to all defined fixed weights on UCF-101 (split 1) as in Table **??**. For the fixed weight case, combined weight of 0.4 and 0.6 for spatial and temporal stream respectively gives the best accuracy. However gating CNN is still better than those pre-defined fixed weights.

When weighting the temporal segment network using our trained gating CNN, it obtains best result and approaches optimum result compared to the case of fixed weights as in Table **??** for UCF-101 split 1. The fixed weight of temporal segment network tends to choose combination 0.5 and 0.5 for spatial and temporal stream (average) as it gives best accuracy result. However our gating network still outperforms fixed weight by margin of 0.24 % which confirms our approach that it needs to weight each sample rather than fixed weights for all samples. We believe that this margin can be improved more with better gating CNN training protocol for future work. Table **??** shows result for HMDB-51 on split 1, it shows improvement compared to the best result of fixed weight (0.5 and 0.5 for spatial and temporal stream respectively) with margin of 0.07 %. HMDB-51 has fewer training data than those

TABLE 5.4: ResNet 40 epochs on UCF-101

RGB weight	Flow weight	Test Accuracy
1.0	0.0	69.83 %
0.0	1.0	63.84%
0.9	0.1	71.60 %
0.8	0.2	72.98 %
0.7	0.3	74.36 %
0.6	0.4	75.76 %
0.5	0.5	76.39 %
0.4	0.6	76.39 %
0.3	0.7	74.94 %
0.2	0.8	72.24 %
0.1	0.9	69.49 %
gating VGG-16		**77.21** %
gating ResNet-50		74.36 %
gating classifier		**77.21** %

TABLE 5.5: ResNet 80 epochs on UCF-101

RGB weight	Flow weight	Test Accuracy
1.0	0.0	70.47 %
0.0	1.0	64.82%
0.9	0.1	72.21 %
0.8	0.2	74.06 %
0.7	0.3	75.49%
0.6	0.4	76.78 %
0.5	0.5	77.50 %
0.4	0.6	77.11 %
0.3	0.7	76.15 %
0.2	0.8	73.40 %
0.1	0.9	70.55 %
gating VGG-16		**78.11** %
gating ResNet-50		75.50 %
gating classifier		77.80 %

TABLE 5.6: ResNet 300 epochs (already overfit) on UCF-101 (split 1)

RGB weight	Flow weight	Test Accuracy
1.0	0.0	78.08 %
0.0	1.0	74.89%
0.9	0.1	80.10 %
0.8	0.2	82.08 %
0.7	0.3	83.64 %
0.6	0.4	84.72 %
0.5	0.5	86.22 %
0.4	0.6	86.25 %
0.3	0.7	85.30 %
0.2	0.8	82.97 %
0.1	0.9	79.61 %
gating VGG-16		**86.25** %
gating ResNet-50		83.64 %
gating classifier		85.30 %

of UCF-101 that is challenge for gating network training. Due to that, we observe minor improvements on HMDB-51 results. There still an opportunity to be improved by means multi task learning.

The results for the UCF-101 and HMDB-51 datasets are given in Table **??** and **??** respectively. For the expert networks that we train using Chainer (Tokui et al., 2015) framework, the proposed baseline gating scheme outperforms all other models. Note that, in this testing, we use data augmentation of center crop for both spatial and temporal in this experiment to save computation time. We compare our proposed models that are shown in Figure 1 (model A, model B, and model C). It is found that gated CNN with 0.3% over averaging fusion (model B) while compared to SVM fusion (model C), it exceeds 1.5%. It also improves both RGB and optical flows alone with 10,2% and 6.5% respectively. ResNet-50 expert network (ResNet-50 for expert network and VGG-16 for gating network) gives better result based on our experiment compared to VGG-16 expert network with large margin of 6.1%. Result confirms the mutual information provided by spatial and motion modality. It also shows integration capability of gating CNN. For HMDB-51, it is found

TABLE 5.7: Gated good practice of two streams trained by (Wang et al., 2015) on UCF-101 (split 1)

RGB weight	Flow weight	Test Accuracy
1.0	0.0	79.34 %
0.0	1.0	83.60%
0.9	0.1	82.10 %
0.8	0.2	84.35 %
0.7	0.3	86.47 %
0.6	0.4	88.16 %
0.5	0.5	89.32 %
0.4	0.6	90.02 %
0.3	0.7	89.67 %
0.2	0.8	88.65 %
0.1	0.9	86.73 %
gating		**91** %

that gated CNN is better with 0.5% over averaging fusion. It also improves RGB or optical flows alone with 5% and 12% respectively. Note that for temporal stream, we use 3 consecutive stacked flow fields with number of displacement from one flow field to the next is 2.

Table **??** shows comparison with another fusion methods. Feichtenhofer's fusion method use late fusion with VGGM2048 and VGG-16 with one loss function. With the same VGG-16, RGB itself achieves 82.61 % and Flows achieve 86.25 % while its fusion achieves 90.62 %. Our experiment on the same two stream achieves 91 % with RGB of 79.34 % and Flows of 83.60 % which means while two expert networks are actually weaker, our gating network achieves comparable performance. Another fusion method is feature amplification with multiplication, even without any information about RGB and Flows alone, it achieves 89.1 % in which our result is slightly better with margin of 1.9 %.

By comparing with the state of the arts we can see that gating CNN improves all the expert type either two stream VGG-16 or temporal segment networks as in Table **??** for UCF-101 and HMDB-51. We use weight of trained networks of two stream networks (Wang et al., 2015) that gives the highest

TABLE 5.8: Gated temporal segment network on UCF-101 (split 1)

RGB weight	Flow weight	Test Accuracy
1.0	0.0	85.87 %
0.0	1.0	87.89%
0.9	0.1	89.63 %
0.8	0.2	91.63 %
0.7	0.3	92.98 %
0.6	0.4	93.62 %
0.5	0.5	93.86 %
0.4	0.6	93.66 %
0.3	0.7	93.14 %
0.2	0.8	91.8 %
0.1	0.9	89.98 %
gating		**94.10** %

accuracy according to their experiments. The main concern is comparison with just averaging fusion and SVM fusion, gated two streams achieved better accuracy with difference of 4.8 % and 0.5 4 % with averaging and SVM fusion respectively on UCF-101. When compared to two streams good practice as elaborated in Table **??**, it has better accuracy with margin of 0.8 % over averaging fusion of two streams good practice.

5.3.3 Discussion

We have tried several gating schemes that basically use deep CNN for weighing. It has shown that VGG-16 give most optimum solution compared to the deeper network of ResNet-50 and shallower network. In the middle of training, the simple classifier (2 layers which have 4096 inputs and 1000 outputs) is robust for approaching optimum solution, however, when as the training converges, there is a shift of variance between training and test in which simple classifier does not hold. For deeper network, it tends to have high degree of freedom as the number of layers is high. As in ResNet-50, even though the number of parameters is fewer than VGG-16, with deeper layers (50), it fails to approach optimum solution. Even residual learning of ResNet-50 tends

TABLE 5.9: Gated temporal segment network on HMDB-51 (split 1)

RGB weight	Flow weight	Test Accuracy
1.0	0.0	54.31 %
0.0	1.0	62.35%
0.9	0.1	59.15 %
0.8	0.2	63.46 %
0.7	0.3	66.73 %
0.6	0.4	68.95 %
0.5	0.5	69.93 %
0.4	0.6	69.93 %
0.3	0.7	68.63 %
0.2	0.8	67.45 %
0.1	0.9	65.36 %
gating		**70.00** %

TABLE 5.10: UCF-101 (split 1)

Methods	Accuracy
Spatial streams (3 channels RGB)	72.7 %
Motion streams (3 flow fields)	76.5%
SVM Fusion (model B)	81.5 %
Averaging (model A)	82.7 %
Gating network (model C) VGG-16	83 %
Gating network (model C) ResNet-50	88.5 %

to benefit from the fewer number of parameters if found to be beneficial for classification not for gating. Further work is to investigate ideal model for optimally weighing expert networks.

TABLE 5.11: HMDB-51 (split 1)

Methods	Accuracy
Spatial streams (3 channels RGB)	36 %
Motion streams (3 flow fields)	43%
Averaging (model A)	47.5 %
Gating network (model C)	48 %
Temporal segment network (averaging) (Wang et al., 2016)	69.93 %
Our gating network (model C) + expert network of Temporal segment network Wang et al., 2016	70 %

TABLE 5.12: Comparison with another fusion method

Methods	RGB	Flow	Fusion
Feichtenhofer of late fusion - VGG-M-2048 (Feichtenhofer, Pinz, and Zisserman, 2016)	74.22 %	82.34 %	85.94
Feichtenhofer of late fusion - VGG-16(Feichtenhofer, Pinz, and Zisserman, 2016)	82.61%	86.25 %	90.62
feature amplification + multiplicative (Park et al., 2016)	- %	- %	89.1 %
Our gating VGG-16 + expert streams of (Wang et al., 2015)	79.34 %	83.60%	91%

TABLE 5.13: Comparison with state of the arts

Methods	UCF-101	HMDB-51
Slow fusion spatio temporal (Karpathy et al., 2014)	36 %	36 %
Improved dense trajectories (IDT) (Wang and Schmid, 2013)	85.9%	57.2 %
Two stream (averaging fusion) (Simonyan and Zisserman, 2014a)	86.2 %	-
Two stream (SVM fusion) (Simonyan and Zisserman, 2014a)	87.0 %	-
Two stream of good practice (Wang et al., 2015)	90.2 %	-
Our gating stream + good practice of (Wang et al., 2015) (VGG-16 gating)	91 %	-
Temporal segment network (Wang et al., 2016)	93.86 %	69.93%
Our gating stream + temporal segment network of (Wang et al., 2016)(VGG-16 gating)	94.1 %	70%

Chapter 6

Conclusion

In Chapter 2, We have proposed method for extracting action features from video namely FLAC over time. This framework is based on spatial and orientational flow autocorrelations of local flow fields and derive shift-invariance and additivity as in HLAC,CHLAC, and GLAC. For FLAC. The optical flows are sparsely described in terms of magnitude and orientation. Since the autocorrelation statistic is used, these method extracts local region of flow field over action cycle that extent the use of standard HOF. In experiments for human action recognition, the proposed methods produce comparable results compared with state of the arts. It is turned out to be complementary with spatial binning of HOF and normalisation to capture the action cycles performed by human even the further works remain.

In Chapter 3, We have proposed Wavelet Packet method as contribution. The proposed method is sparse cuboid with multiresolution Haar Wavelet Packet along spatio temporal cuboid. The theoretical foundation is delivered in how to treat dense optical flow using this approach. This feature is embedded with BoF giving it robust to scale variation. The proposed method is better than state of the arts in which mostly motion-based methods. We found that even sampling step is not highly dense as previous works, with bigger cuboid Packet Flow will spatially pool dominant motion by minimising noise and capture detail of temporal dynamics.

Based on experimental results, we have compared with state of the arts

using KTH and UCF Sports dataset and found comparable on KTH and outperforming on UCF Sports. Even though dense optical flow is used as base of our proposed method, it produces efficient computation and reliable performance. Detail temporal dynamics in multi-resolution and BoF manner are potentially giving high degree of freedom in performance, thus trial and error are needed for depth selection. For a small number of classes and high number of samples in which produce high density per class would converge description of temporal dynamics of motions. However, large classes and a little number of samples mean high degree of freedom that will cause uncertainty but still we can discover which resolution is better suited for generalization.

For future works, we can be leveraged into concatenation with another feature such as shape-based descriptors. There is also chance to adapt into another motion-based features such as trajectory level generation or Slow Features Analysis (SFA) to enrich motion characteristic in more detailed form. There are opportunities to improve by making use of optimisation solution, camera motion removal, or localization methods. Furthermore, as the rise of deep neural convolutional networks, it could be used to analyze temporal dynamics to observe its detailed properties for action classifications. Moreover, this method can be adopted in broader motion-based vision topics such as dynamic scene or action based movie understanding.

In Chapter 4, We have demonstrated local motion superpixel evolution over time using three principal local features, namely HOF correlation, centers of mass, and HOG. By tracking the centers of mass of motion superpixels over time, feature vectors form time series data that can be used to analyze temporal dynamics. Moreover, superpixels capture locality evolution for motion that is important for achieving significant video classification performance. To enrich the temporal information, various sizes of superpixels, spatial and motion descriptors, and two-way tracking with separate BOFs

can be applied. We have applied our approach to the UCF Sports dataset and the JPL First-Person Interaction dataset and found it to be comparable with existing methods. Future research will involve concatenation with global motion superpixels. There is also a possibility of merging these ideas with CNNs to better understand locality and its temporal evolution for various tasks, in particular video classification.

In Chapter 5, We have proposed a baseline gating scheme that able to weigh expert streams for video activity recognition. In this research, the gating CNN is trained to decide which network stream is more salient compared to the others adaptively. To this end, independent loss function and backpropagation are applied for each expert and gating stream, The outputs from the expert streams are then weighted adaptively by gating CNN for each sample.

We have conducted experiments on UCF-101 dataset and HMDB-51 dataset using VGG-16 and resNet-50 to evaluate how deep networks have the ability of expert selection for each sample rather than fixed weights. Results show state of the art performance is achieved when compared to another fusion method. However, gating CNN is burdened from high parameters and degree of freedom while simple classifier tends to overfit with training data. Therefore, further investigation is required to find ideal structure of gating CNN and possible regularization method to overcome aforementioned problems. The gating CNN is potentially useful for various expert networks' integration such as multimodal, multiresolution, source or multisegment along spatiotemporal space. Thus, rather than two modalities, even greater challenge is, whether gating CNN optimally weigh multiple modalities while considering the diversity of sources.

Bibliography

Ahmed, Ejaz, Michael Jones, and Tim K Marks (2015). "An improved deep learning architecture for person re-identification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3908–3916.

Bachem, Olivier et al. (2016). "Fast and provably good seedings for k-means". In: *Advances in Neural Information Processing Systems*, pp. 55–63.

Bergh, Michael Van den et al. (2012). "Seeds: Superpixels extracted via energy-driven sampling". In: *European conference on computer vision*. Springer, pp. 13–26.

Bergh, Michael Van den et al. (2013). "Online video seeds for temporal window objectness". In: *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, pp. 377–384.

Bhattacharya, Subhabrata et al. (2014). "Recognition of complex events: Exploiting temporal dynamics between underlying concepts". In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, pp. 2243–2250.

Boughorbel, Sabri, J-P Tarel, and Nozha Boujemaa (2005). "Generalized histogram intersection kernel for image recognition". In: *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. Vol. 3. IEEE, pp. III–161.

Bruna, Joan and Stéphane Mallat (2013). "Invariant scattering convolution networks". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1872–1886.

Byrne, Jeffrey (2015). "Nested motion descriptors." In: *CVPR*, pp. 502–510.

Chakraborty, Bhaskar et al. (2012). "Selective spatio-temporal interest points". In: *Computer Vision and Image Understanding* 116.3, pp. 396–410.

Chen, Quan-Qi and Yu-Jin Zhang (2016). "Cluster trees of improved trajectories for action recognition". In: *Neurocomputing* 173, pp. 364–372.

Dauphin, Yann N et al. (2016). "Language modeling with gated convolutional networks". In: *arXiv preprint arXiv:1612.08083*.

Dong, Xuan, Ah-Chung Tsoi, and Sio-Long Lo (2014). "Superpixel appearance and motion descriptors for action recognition". In: *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, pp. 1173–1178.

Farnebäck, Gunnar (2003). "Two-frame motion estimation based on polynomial expansion". In: *Scandinavian conference on Image analysis*. Springer, pp. 363–370.

Fathi, Alireza and Greg Mori (2008). "Action recognition by learning mid-level motion features". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, pp. 1–8.

Feichtenhofer, Christoph, Axel Pinz, and AP Zisserman (2016). "Convolutional two-stream network fusion for video action recognition". In:

Fernando, Basura et al. (2017). "Rank pooling for action recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 39.4, pp. 773–787.

Fleet, David J and Allan D Jepson (1990). "Computation of component image velocity from local phase information". In: *International journal of computer vision* 5.1, pp. 77–104.

Gers, Felix A, Jürgen Schmidhuber, and Fred Cummins (1999). "Learning to forget: Continual prediction with LSTM". In:

Gokhale, MY and Daljeet Kaur Khanduja (2010). "Time domain signal analysis using wavelet packet decomposition approach". In: *International Journal of Communications, Network and System Sciences* 3.03, p. 321.

Hadjidemetriou, Efstathios, Michael D Grossberg, and Shree K Nayar (2004). "Multiresolution histograms and their use for recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.7, pp. 831–847.

Hadsell, Raia, Sumit Chopra, and Yann LeCun (2006). "Dimensionality reduction by learning an invariant mapping". In: *Computer vision and pattern recognition, 2006 IEEE computer society conference on*. Vol. 2. IEEE, pp. 1735–1742.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Jain, Mihir, Herve Jegou, and Patrick Bouthemy (2013). "Better exploiting motion for better action recognition". In: *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, pp. 2555–2562.

Ji, Shuiwang et al. (2013). "3D convolutional neural networks for human action recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 35.1, pp. 221–231.

Jiang, YG et al. (2014). *THUMOS challenge: Action recognition with a large number of classes*.

Karpathy, Andrej et al. (2014). "Large-scale video classification with convolutional neural networks". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732.

Ke, Yan, Rahul Sukthankar, and Martial Hebert (2007). "Spatio-temporal shape and flow correlation for action recognition". In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, pp. 1–8.

Kendall, Alex, Yarin Gal, and Roberto Cipolla (2017). "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics". In: *arXiv preprint arXiv:1705.07115*.

Khan, Fahad Shahbaz, Joost Van de Weijer, and Maria Vanrell (2012). "Modulating shape features by color attention for object recognition". In: *International Journal of Computer Vision* 98.1, pp. 49–64.

Kim, Gyu-Jin et al. (2010). "Automated measurement of crowd density based on edge detection and optical flow". In: *Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on*. Vol. 2. IEEE, pp. 553–556.

Klaser, Alexander, Marcin Marszałek, and Cordelia Schmid (2008). "A spatio-temporal descriptor based on 3d-gradients". In: *BMVC 2008-19th British Machine Vision Conference*. British Machine Vision Association, pp. 275–1.

Kobayashi, Takumi and Nobuyuki Otsu (2008). "Image feature extraction using gradient local auto-correlations". In: *European conference on computer vision*. Springer, pp. 346–358.

Laine, Andrew and Jian Fan (1993). "Texture classification by wavelet packet signatures". In: *IEEE Transactions on pattern analysis and machine intelligence* 15.11, pp. 1186–1191.

Lan, Tian, Yang Wang, and Greg Mori (2011). "Discriminative figure-centric models for joint action localization and recognition". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, pp. 2003–2010.

Lan, Zhenzhong et al. (2015). "Long-short term motion feature for action classification and retrieval". In: *arXiv preprint arXiv:1502.04132*.

Laptev, Ivan (2005). "On space-time interest points". In: *International journal of computer vision* 64.2-3, pp. 107–123.

Laptev, Ivan et al. (2008). "Learning realistic human actions from movies". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, pp. 1–8.

Le, Quoc V et al. (2011). "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis". In:

Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, pp. 3361–3368.

Lee, Jehee and Sung Yong Shin (2000). "Multiresolution motion analysis with applications". In: *In Proc. International Workshop on Human Modeling and Animation*. Citeseer, pp. 131–143.

Legenstein, Robert, Niko Wilbert, and Laurenz Wiskott (2010). "Reinforcement learning on slow features of high-dimensional input streams". In: *PLoS Computational Biology* 6.8, e1000894.

Lin, Zhe, Zhuolin Jiang, and Larry S Davis (2009). "Recognizing actions by shape-motion prototype trees". In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, pp. 444–451.

Liu, Ce, Jenny Yuen, and Antonio Torralba (2016). "Sift flow: Dense correspondence across scenes and its applications". In: *Dense Image Correspondences for Computer Vision*. Springer, pp. 15–49.

Lowe, David G (1999). "Object recognition from local scale-invariant features". In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee, pp. 1150–1157.

Lucas, Bruce D, Takeo Kanade, et al. (1981). "An iterative image registration technique with an application to stereo vision". In:

Matsukawa, Tetsu and Takio Kurita (2010). "Action Recognition Using Three-Way Cross-Correlations Feature of Local Moton Attributes". In: *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, pp. 1731–1734.

Mikolajczyk, Krystian and Hirofumi Uemura (2011). "Action recognition with appearance–motion features and fast search trees". In: *Computer Vision and Image Understanding* 115.3, pp. 426–438.

Oshin, Olusegun, Andrew Gilbert, and Richard Bowden (2014). "Capturing relative motion and finding modes for action recognition in the wild". In: *Computer Vision and Image Understanding* 125, pp. 155–171.

Otsu, Nobuyuki and Takio Kurita (1988). "A New Scheme for Practical Flexible and Intelligent Vision Systems." In: *MVA*, pp. 431–435.

Park, Eunbyung et al. (2016). "Combining multiple sources of knowledge in deep cnns for action recognition". In: *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, pp. 1–8.

Rodriguez, Mikel D, Javed Ahmed, and Mubarak Shah (2008). "Action mach a spatio-temporal maximum average correlation height filter for action recognition". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, pp. 1–8.

Sadanand, Sreemananth and Jason J Corso (2012). "Action bank: A high-level representation of activity in video". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp. 1234–1241.

Schindler, Konrad and Luc Van Gool (2008). "Action snippets: How many frames does human action recognition require?" In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, pp. 1–8.

Schuldt, Christian, Ivan Laptev, and Barbara Caputo (2004). "Recognizing human actions: a local SVM approach". In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 3. IEEE, pp. 32–36.

Sermanet, Pierre et al. (2013). "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *arXiv preprint arXiv:1312.6229*.

Shannon, Benjamin J and Kuldip K Paliwal (2006). "Feature extraction from higher-lag autocorrelation coefficients for robust speech recognition". In: *Speech Communication* 48.11, pp. 1458–1485.

Shi, Feng, Emil Petriu, and Robert Laganiere (2013). "Sampling strategies for real-time action recognition". In: *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, pp. 2595–2602.

Shiraki, Takayoshi et al. (2006). "Real-time motion recognition using chlac features and cluster computing". In: *Proceedings of the 3rd IFIP international conference on network and parallel computing*. Citeseer, pp. 50–56.

Simonyan, Karen and Andrew Zisserman (2014a). "Two-stream convolutional networks for action recognition in videos". In: *Advances in neural information processing systems*, pp. 568–576.

— (2014b). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Somasundaram, Guruprasad et al. (2014). "Action recognition using global spatio-temporal features derived from sparse representations". In: *Computer Vision and Image Understanding* 123, pp. 1–13.

Soomro, Khurram and Amir R Zamir (2014). "Action recognition in realistic sports videos". In: *Computer Vision in Sports*. Springer, pp. 181–208.

Stein, Barry E, Terrence R Stanford, and Benjamin A Rowland (2009). "The neural basis of multisensory integration in the midbrain: its organization and maturation". In: *Hearing research* 258.1-2, pp. 4–15.

Sun, Lin et al. (2014). "DL-SFA: deeply-learned slow feature analysis for action recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2632.

Sundberg, Patrik et al. (2011). "Occlusion boundary detection and figure/ground assignment from optical flow". In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, pp. 2233–2240.

Ta, Anh-Phuong et al. (2010). "Recognizing and localizing individual activities through graph matching". In: *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*. IEEE, pp. 196–203.

Taylor, Graham W et al. (2010). "Convolutional learning of spatio-temporal features". In: *European conference on computer vision*. Springer, pp. 140–153.

Theriault, Christian, Nicolas Thome, and Matthieu Cord (2013). "Dynamic scene classification: Learning motion descriptors with slow features analysis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2603–2610.

Tokui, Seiya et al. (2015). "Chainer: a next-generation open source framework for deep learning". In: *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*. Vol. 5.

Uijlings, Jasper RR et al. (2014). "Realtime video classification using dense hof/hog". In: *Proceedings of International Conference on Multimedia Retrieval*. ACM, p. 145.

Wang, Heng and Cordelia Schmid (2013). "Action recognition with improved trajectories". In: *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, pp. 3551–3558.

Wang, Heng et al. (2011). "Action recognition by dense trajectories". In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, pp. 3169–3176.

Wang, Limin, Yu Qiao, and Xiaoou Tang (2016). "MoFAP: A multi-level representation for action recognition". In: *International Journal of Computer Vision* 119.3, pp. 254–271.

Wang, Limin et al. (2015). "Towards good practices for very deep two-stream convnets". In: *arXiv preprint arXiv:1507.02159*.

Wang, Limin et al. (2016). "Temporal segment networks: Towards good practices for deep action recognition". In: *European Conference on Computer Vision*. Springer, pp. 20–36.

Wong, Shu-Fai and Roberto Cipolla (2007). "Extracting spatiotemporal interest points using global information". In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, pp. 1–8.

Yu, Weichuan, Gerald Sommer, and Kostas Daniilidis (2003). "Multiple motion analysis: in spatial or in spectral domain?" In: *Computer Vision and Image Understanding* 90.2, pp. 129–152.

Yudistira, Novanto and Takio Kurita (2015). "Multiresolution Local Autocorrelation of Optical Flows over time for Action Recognition". In: *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. IEEE, pp. 1930–1935.

Zeng, Xingyu et al. (2016). "Gated bi-directional cnn for object detection". In: *European Conference on Computer Vision*. Springer, pp. 354–369.