12-15-2019

On Action Quality Assessment

Paritosh Parmar

ON ACTION QUALITY ASSESSMENT

By

Paritosh Parmar

Bachelor of Technology - Electronics and Communication Engineering
Nirma University
2012

Master of Technology - Robotics
SRM University
2014

A dissertation submitted in partial fulfillment
of the requirements for the

Doctor of Philosophy - Electrical Engineering

Department of Electrical and Computer Engineering
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
December 2019

UNLV | GRADUATE COLLEGE

Dissertation Approval

The Graduate College
The University of Nevada, Las Vegas

October 2nd, 2019

This dissertation prepared by

Paritosh Parmar

entitled

On Action Quality Assessment

is approved in partial fulfillment of the requirements for the degree of

Doctor of Philosophy - Electrical Engineering
Department of Electrical and Computer Engineering

Brendan Morris, Ph.D.
Examination Committee Chair

Kathryn Hausbeck Korgan, Ph.D.
Graduate College Dean

Emma Regentova, Ph.D.
Examination Committee Member

Venkatesan Muthukumar, Ph.D.
Examination Committee Member

Mohamed Trabia, Ph.D.
Graduate College Faculty Representative

ABSTRACT

In this dissertation, we tackle the task of quantifying the quality of actions, i.e., how well an action was performed using computer vision. Existing methods used human body pose-based features to express the quality contained in an action sample. Human body pose estimation in actions such as sports actions, like diving and gymnastic vault, is particularly challenging, since the athletes undergo convoluted transformations while performing their routines. Moreover, pose-based features do not take into account visual cues such as water splash in diving. Visual cues are taken into account by human judges. In our first work, we show that using visual representation – spatiotemporal features computed using a 3D convolutional neural network – is more suitable as those attend to appearance and salient motion patterns of the athlete's performance. Alongwith developing three action quality assessment (AQA) frameworks, we also compile a diving and gymnastic vault dataset. Rather, learning an action-specific model, in our second work, we show that learning to assess the quality of multiple actions jointly is more efficient as it can exploit shared/common elements of quality among different actions. All-action modeling better uses the data, shows better generalization, and adaptation to unseen/novel action classes. Taking inspiration from the 'learning by teaching' method, we propose to take multitask learning (MTL) approach to AQA, unlike existing approaches, which follow single task learning (STL) paradigm. In our MTL approach we force the network to delineate the action sample – recognize the action in detail, and commentate on good and bad points of the

performance, in addition to the main task of AQA scoring. Through this better characterization of action sample, we are able to obtain state-of-the-art results on the task of AQA. To enable our MTL approach, we also released the largest multitask AQA dataset, MTL-AQA. Additionally, in the interest of readers, we have included an introductory chapter and reviewed related work.

TABLE OF CONTENTS

ABSTRACT iii

LIST OF TABLES viii

LIST OF FIGURES ix

CHAPTER 1 INTRODUCTION 1
Action Quality Assessment vs. Action Recognition 4
Factors of quality in actions 7
Outline 9

CHAPTER 2 LITERATURE REVIEW 12
Action Quality Assessment 12
Skills Assessment 22

CHAPTER 3 LEARNING TO SCORE OLYMPIC EVENTS 27
Introduction 27
Approach 28
Computing spatiotemporal features 29
Frameworks 30
Aggregation schemes 30
Regression models 31
Datasets 32
Experiments 34
Conclusion 37

CHAPTER 4 ACTION QUALITY ASSESSMENT ACROSS MULTIPLE ACTIONS 38
Introduction 38
AQA-7 Dataset 39
Common Action Quality Elements 43
Approach 43
Experiments 45
Conclusion 49

CHAPTER 5 MULTITASK LEARNING APPROACH TO ACTION QUALITY ASSESSMENT 54
Introduction 54
Multitask Approach to AQA 55
Multitask AQA Dataset 59
Experiments 61
Discussion 65

CHAPTER 6 CONCLUSION 66

BIBLIOGRAPHY 73

CURRICULUM VITAE 74

LIST OF TABLES

2.1 Summary of AQA works. 21
2.2 Summary of movement analysis works. 22
2.3 Summary of Skills Assessment works. 26

3.1 Frameworks. 32
3.2 Performance comparison on Diving action. 35
3.3 Performance comparison on Gymnastic Vault action. 36
3.4 Performance comparison on Figure Skating action. 36

4.1 Characteristics of AQA-7 dataset. 42
4.2 All-Action vs. Single-Action models. Performance evaluation of single-action and all-action models in terms of action-wise and average Spearman's rank correlation (higher is better). First two frameworks simply average features to aggregate them and use SVR as the regression module. The bottom two frameworks use LSTM to aggregate features and use a fully-connected layer as the regression module. Our approach can be directly compared with single-action C3D-LSTM [28], since both have the same architecture. 50
4.3 Zero-shot AQA. Performance comparison of randomly-initialized model, single-action models (for , first row shows the results of training on diving action measuring the quality of the remaining (unseen) action classes), and multi-action model (all-action model trained on five action classes) on unseen action classes. In multi-action class, the model is trained on five action classes and tested on the remaining action class (column-wise). In single-action model rows, diagonal entries show results of training and testing on the same action. Avg. Corr. shows the result of average (using Fisher's z-score) correlation across all columns. 51
4.4 Finetuning from scratch vs. finetuning from pre-trained multi-action model. Experimental results (Spearman's rank correlation) of finetuning a randomly-initialized (RI) model and an all-action (AA) model pre-trained on five action classes. The numbers represent the best results from all the iterations. 52

5.1 MSCADC-MTL architecture. C3(d,ch): 3D convolutions, ch-no. of channels, d-dilation rate. C1: 1x1x1 convolutions. BN: batch normalization. MP(kr): max pooling operation, kr-kernel size. Cntxt net: context net for multi-scale context aggregation. AP: average pooling across (2x11x11) volume. 58
5.2 Details of our newly introduced dataset, and its comparison with the existing AQA datasets. 60

5.3 Classification of dives. Each combination of the presented sub-fields produces a different kind of maneuver. 61
5.4 STL vs. MTL across different architectures. Cls - classifiction, Caps - captioning. First row shows STL results, while the remaining rows show MTL results. 62
5.5 Performance comparison with the existing AQA approaches. 63
5.6 Performance on auxiliary tasks. Comparison with [23] on the task of dive classification is presented in the top table. Bleu(B), Meteor(M), Rouge(R), CIDEr(C) scores for the captioning task are presented in the bottom table. 63
5.7 STL vs. MTL generalization. Training using increasingly reduced no. of training samples. 64
5.8 Performance of fitting linear regressors on the activations of all the convolutional layers. 65

LIST OF FIGURES

1.1 Application in phyiostherapy to evaluate if the patient is doing exercises correctly. Left - erroneous execution, Right - correct execution. 2
1.2 AQA applied in Olympic sports sector. Final Predicted Score 16/20, which implies that action quality score is good 3
1.3 Assessing surgical skills of a med student. 5
1.4 Definition of AQA vs. action recognition. 5
1.5 Amount of evidence needed to carry out the tasks meaningully. 6
1.6 Thesis summary 11

3.1 Frameworks. 32

4.1 Preview of AQA-7 dataset. 41
4.2 Illustration of common action quality elements. 44
4.3 Finetuning from scratch vs. finetuning from pre-trained multi-action model. Plot of Spearman's rank correlation against every hundred iterations for different number of training samples. Blue and red curves represent multi-action and randomly initialized models, respectively. The gap in the initial iterations suggest that good initialization of LSTM weights was achieved by training on multiple actions. In most of the cases, multi-action model has better performance than randomly initialized model on test samples throughout all the iterations. 53

5.1 Architecture of C3D-AVG-MTL. 57

CHAPTER 1

INTRODUCTION

How well did you dive? Out of 100 points, how many points would you get for your dive? We can ask similar questions for any other person doing other actions like gymnastic vault, figure skating, etc. Essentially, in these questions, we are asking *how well* the action was performed by the person. Quantifying *how well* an action was performed can be refered to as *assessing the quality of actions* or *action quality assessment*, AQA, for short.

Quality of almost any action can be measured, for *e.g.*, we can assess how well a golfer can swing or how well an athlete is doing their physical rehabilitation exercises, etc. Therefore, AQA has applications in many fields like:

- Physical rehabilitation program: During physical rehabilitation, patients are required to do exercises, which will enable patients to regain the mobility or the conditioning. Physiotherapists monitor and assess *how* the patients are doing these recommended exercises. Consulting physiotherapists may not be a feasible option for financially struggling demographic. Automated, at-home, low-cost physical rehabilitation option can be provided using automated assessment of exercises (which are essentially actions) using computer vision.

- Automated Olympics judging: Biasing and scandals involving partial judging is not a

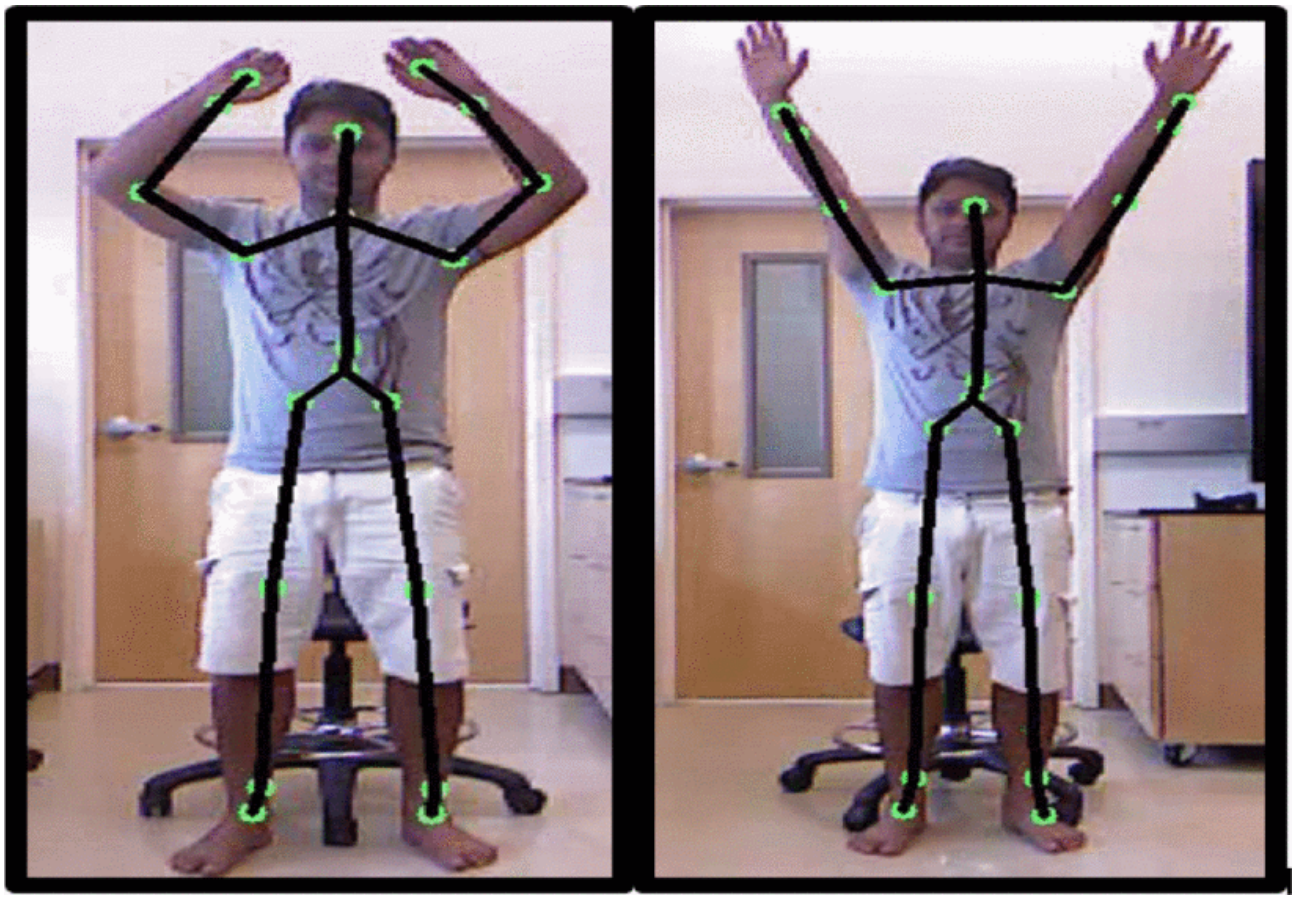


Figure 1.1: Application in phyiostherapy to evaluate if the patient is doing exercises correctly. Left - erroneous execution, Right - correct execution.

Figure 1.2: AQA applied in Olympic sports sector. Final Predicted Score 16/20, which implies that action quality score is good

new thing in sports judging. An automated sports judging computer vision system can be used to provide a second opinion in case of a controversial decision. This kind of system can also be used to detect if judging was partial. Access to high-level coaching might not be an option for a larger part of the society. Automated action quality assessing system can be used to act as a judge for performances (diving, gymnastics, skiing, etc.), and also provide feedback like a coach.

- Assessing skills: There's always a demand for skilled labor. But honing those skills re-

quires hours and hours of practice and feedback. Automated vision-based skills assessment systems can very handy in these situations. Such a system can be used to monitor the performance and progress of a person practicing to develop the skills, and also provide feedback in case of erroneous execution. These systems can be useful in assessing skills of everyday activities (drawing, painting, applying make-up, etc.), and specialized activities (surgical skills, pottery, woodworking, spray painting, etc.).

Action Quality Assessment vs. Action Recognition

In order to better understand AQA, and to gain more insights into the challenges that we face in AQA, in the following, we compare and contrast AQA with a closely related subfield of computer vision, action recognition.

Definition of tasks:

AQA: task is quantify how well an action performed in a given action instance.

Action recognition: task is to identify what action was performed in a given action instance.

Intrasample differences:

AQA: Since the samples are from the same action class, the differences among instances are subtle.

Action recognition: Since the instances are from different classes, the differences among instances are significant.

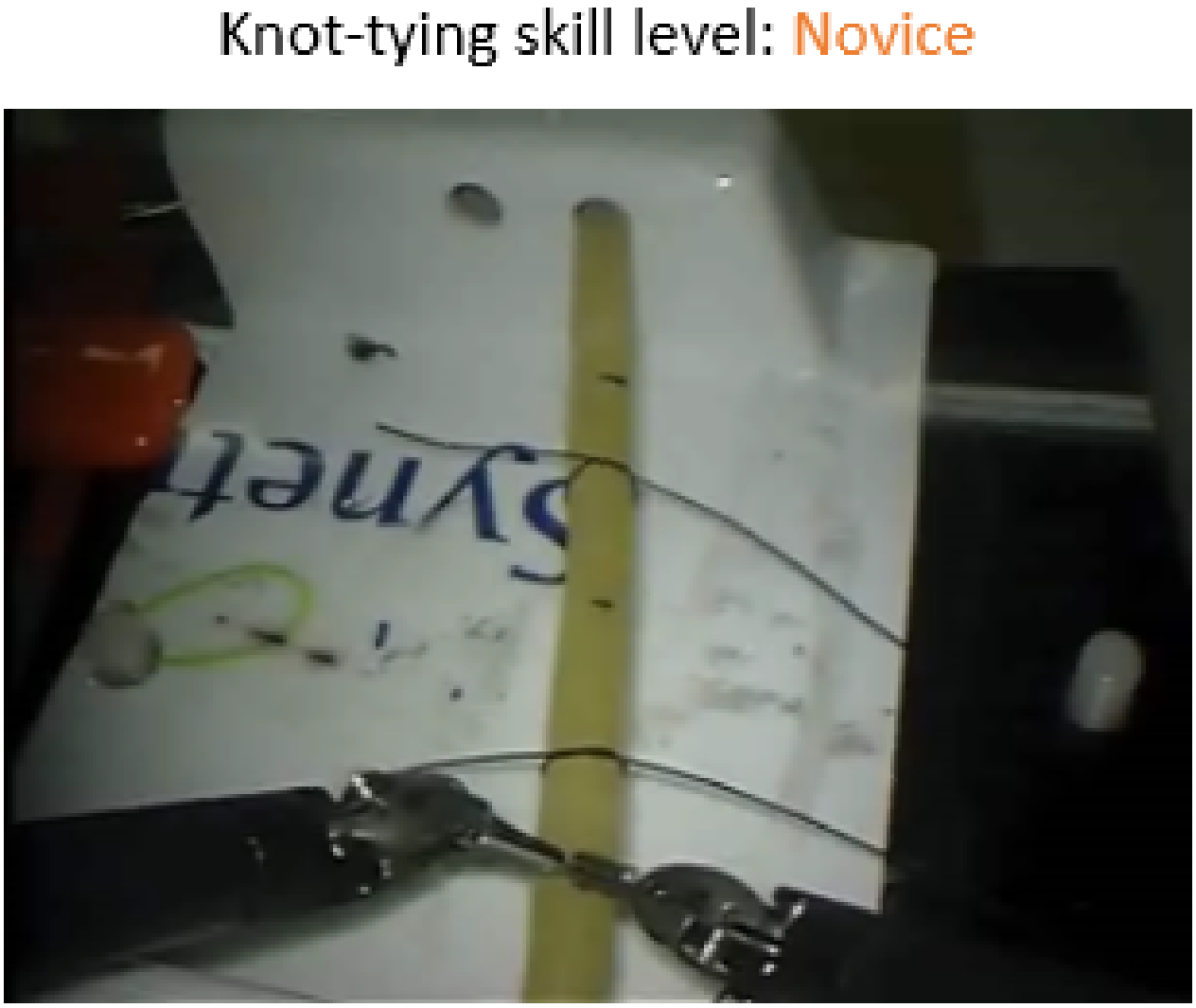


Figure 1.3: Assessing surgical skills of a med student.

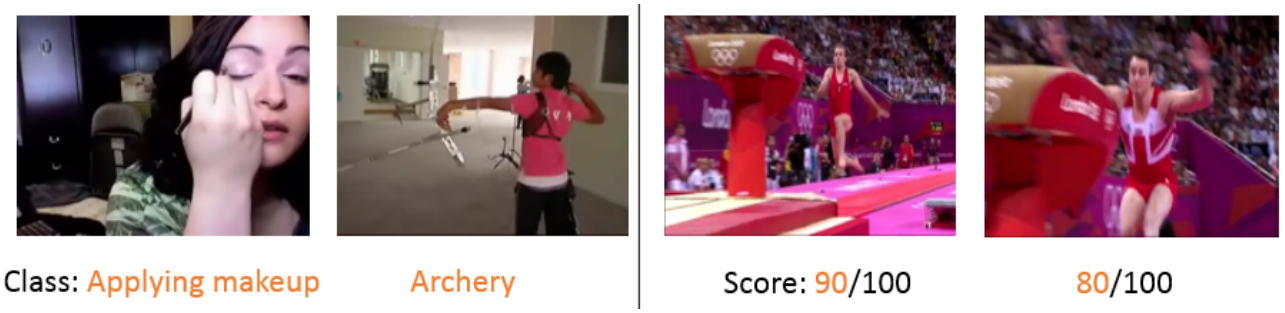


Figure 1.4: Definition of AQA vs. action recognition.

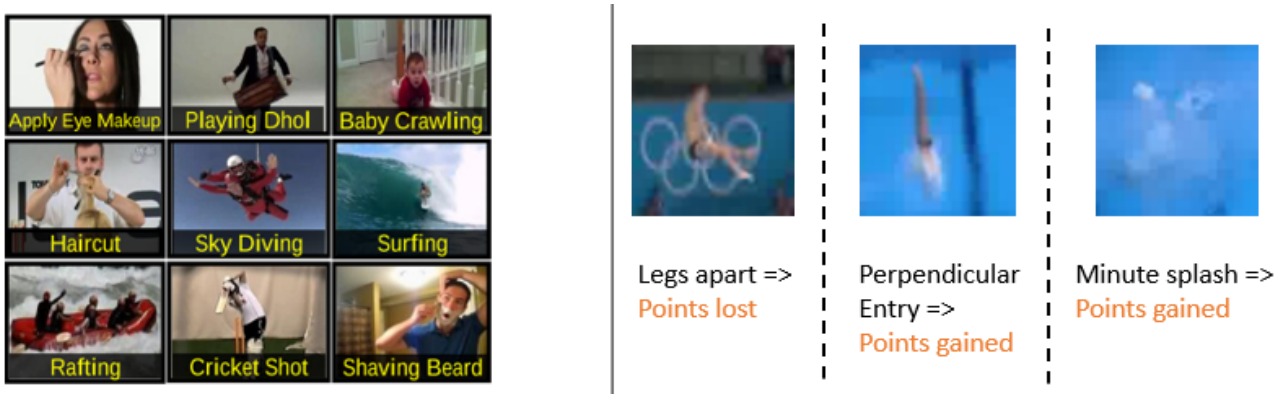


Figure 1.5: Amount of evidence needed to carry out the tasks meaningully.

Amount of evidence to consider:

AQA: Whole action sequence needs to be "seen" in order to correctly and meaningfully quantify the quality of action. It won't be meaningful to predict the quality of an action from a small segment of the action sequence, because the performer might make an error at any time during the course of action. For example, let's take diving action. The diver might be perfectly executing the dive in the air, but might make an error while entering the water, or the diver might be not-so-perfect in the air, but somehow manage to make a good entry. So, if we were to judge the quality of action from just a short action clip while the diver was in air, it would be poor assessment of the action quality because error made by the diver during entry into the water was not taken into account. However, if the whole action sequence was considered/"seen", then the diver would be penalized for erroneous entry into the water.

Action recognition: It has been shown that an action can be correctly classified from less evidence as just a single video frame. Generally, due to significant differences among action classes, action can be classified after seeing much lesser evidence.

Dataset sizes: Dataset size is a crucial factor in a system's performance.

AQA: AQA datasets are very small compared to action recognition datasets. Dataset sample sizes are generally, a couple of hundred samples. For *e.g.*, diving - 370 samples; gymnastic vault - 176 samples; figure skating - 170 samples.

Action recognition: Action recognition datasets are pretty large, for *e.g.*, UCF101 [34] - 13000 samples; Kinetics [5]; Sports1M [16] - 1 million samples; YouTube8M [3] - 8 million samples.

Factors of quality in actions

In this section we discuss, using examples, some of the positive and negative factors of quality in actions.

Case: Olympic events

Body position: Athletes are required to perform routines which consist of, for example, twisting and somersaulting in air. They are required to do so while having their bodies in different positions, like in a tuck position or a pike position. Twisting and somersaulting is especially more difficult to perform while maintaining bodies in *tight* tuck or pike position. So, how tightly an athlete is maintaining their forms is used a performance metric to separate higher-level athletes from lower-level athletes.

Keeping legs straight in pike position is another condition that serves as a performance metric that is a factor in determining the final action quality score.

Splash: In diving, water splash made when the athlete enters water is another factor in

determining the final score. In order to minimize the splash, the athlete needs to enter with their body perfectly vertical. More their body is bent or at an angle, larger would be the splash. To have a vertical body at the time of entry, the athlete needs to get through all the maneuvers like twists and somersaults at the correct time, which requires ideal execution and good skill set.

Landing: In sports like Gymnastics, Skiing, and Snowboarding, equivalent of splash is landing on the mat or ground. Similar to like in diving, the athlete needs to get through all the twists, somersaults, and spins in order to make a good, stable landing, and not stepping outside the landing area limits, which is hard to accomplish, and therefore plays an important factor in the final action quality score.

Case: Exercises

Stretching in physiotherapy: In many cases, people develop less mobility (temporary or permanent) due to accidents or diseases like cerebral palsy. Their mobility issues can be improved or managed through regular exercises. In these exercises, patients are asked to stretch their limbs or other body parts in a particular way. If the patient is able to stretch all the way, that is considered to be a success. In case they are not able to stretch all the way, more they are closer to the ideal form, the better progress it is for the patients.

Form in weight training: Maintaining form and isolation are important factors when doing weight training. If the person doing exercises is using/taking aid from unintended muscle groups in order to lift weights, it would be considered a negative aspect, and should be penalized when quantifying the quality of weight training. Where as if the person was able to perform

the weight lifts using only the target muscle groups, it would be considered a positive aspect.

Case: Surgical skill

Respect for the tissue: If the surgeon is frequently using unnecessary force on tissue while handling, the excessive force can cause damage to the tissue. Likewise, if the surgeon is not careful they can damage the tissue with inappropriate use of tools. These factors would be considered negative factors. Where as if the surgeon was careful in handling and paid attention to not cause any damage to the tissue, it would be considered as a positive aspect.

Time and motion: If the surgeon is not able to figure out optimum way during the surgical process, they may waste time in unnecessary moves which is viewed as negative aspect, because, for example, taking more time, can mean result into more blood loss. However, if the surgeon had a good economy of movement and efficiency, those factors would be considered positive.

Flow of operation: Surgical process can be made smooth through planning. If the surgeon had not planned beforehand the movement may be jittery due to hesitation and lack of planning. This would be considered negative. Effortless flow, on the other hand, would be considered a positive aspect.

Outline

AQA can be defined as a function of what action was performed and how well that action was performed.

$$AQA = \int (what\ you\ performed, how\ well\ you\ performed) \tag{1.1}$$

Improving the performance on the task of AQA translates to better estimating or finding/developing f in Eq. 1.1. In this thesis, we present computer vision based approaches to assessing the quality of actions. Particularly, we propose to learn and use better representations that help learn estimating function, f. The thesis can be summarized as in Fig. 1.6. In the following, we discuss the main contributions of our works.

- In Chapter 3, we focus on using better visual representations of actions and treat f as a regressor as opposed to a classifier as in [27]. Previous works [31, 39] use human pose features for AQA. Poor pose estimation in sports domain adversely affects assessment of action quality. To this end, we propose to use spatiotemporal features learnt using a 3D CNN. Using visual information directly, allows us to heed to important visual cues like splash in Diving. We propose three frameworks, all of which surpass previously best recorded performances. Since then, numerous state-of-the-art approaches [41, 20, 21, 42, 8] have adopted to use visual information directly.

- To exploit common/shared action quality elements across different actions, in Chapter 4 we continue to treat f as a regressor, but propose to learn it across multiple actions. Through experimentation, we are able to show that by exploiting common action quality elements we are able to make better usage of available data, and gain better generalization and adaptability to unseen action classes.

- In Chapter 5, we note take a note that f is in fact a function of functions, and propose to learn f by optimizing its functions. By learning to delineate an action through learning detailed action classification and detailed commentary generation, we are able to

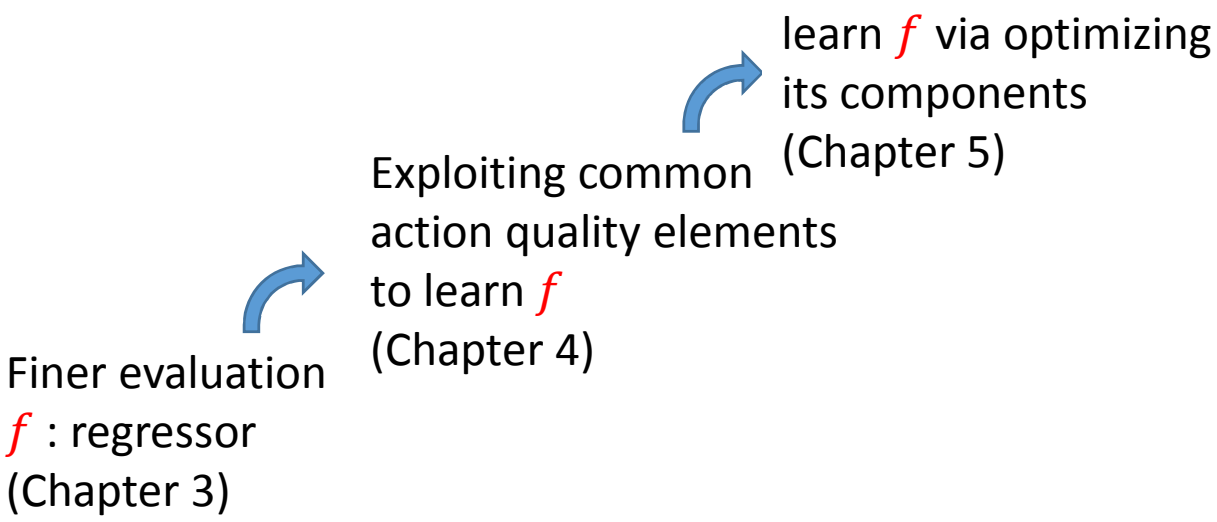


Figure 1.6: Thesis summary

achieve new state-of-the-art results. In addition, we design multitask architectures that are trainable end-to-end. We show that, multitask learning approach outperforms single task learning because of the better characterization and generalization.

Since, AQA is an upcoming sub-field, there is a shortage of datasets. To this end, through our works, we have compiled and publicly released the following datasets:

- UNLV-Dive; UNLV-Gymvault
- AQA-7: largest AQA dataset covering seven actions
- MTL-AQA: first multitask AQA dataset

CHAPTER 2

LITERATURE REVIEW

AQA, unlike other action related tasks such as action recognition and action detection, is relatively newer field and has received lesser attention despite having wide applications. We had started working on AQA in the year 2015; until then very few efforts have been toward AQA. Although, AQA seems to have started gaining more attention and more efforts have been put toward AQA since the year 2017. In the following, we review the AQA-oriented works before we started working on this thesis and also the works since we started working towards this thesis.

Action Quality Assessment

In their pioneering work, Pirsiavash *et al.* [31], formulated the problem of assessing the quality in the area of Olympic judging. They released first Olympic judging dataset comprising of short length and long length actions, Diving, and Figure Skating, respectively. They proposed to estimate pose of athlete in every frame using the method developed by Yang *et al.* [43]. Frame-level pose features are concatenated to form a long sample-level representation. These long features are post-processed using DCT/DFT to reduce their dimensionality and get rid of high-frequency noise. A SVR is then used to map these post-processed features to AQA scores. Apart from pose-based features, they also experiment with spatio-temporal interest points and

hierarchical convolutional features. While pose-based features worked best Diving, hierarchical features worked best for Figure Skating.

Another pose-based approach was developed by Venkataraman *et al.* [39], calculate the approximate entropy of pose features (which better encodes dynamical information than DCT/DFT), which are then concatenated.

Pose estimation is difficult in sports domain. Pose-only features do not take into account visual cues like splash in Diving. Parmar *et al.* [28] hypothesized spatio-temporal features might work better on AQA task. So, they proposed to use C3D features. They also released new dataset of Gymnastic Vault action, and extended Diving dataset. Their proposed frameworks worked better all actions – Diving, Gymnastic Vault, Figure Skating.

Instead of averaging the features uniformly like Parmar *et al.* [28], Xiang *et al.* [41]proposed to fuse action segment-specific features. Segment-specific averaging helped push further the performance on assessing the quality of Diving. In addition to using segment aware fusion, Li *et al.* [20] proposed to use ranking loss in addition to generally used Euclidean loss. This along with some network modification helped improve the performance on both Diivng and Gymnastic Vault datasets.

Unlike previously discussed approaches, which took action-specific modeling approach, Parmar *et al.* [26] hypothesized that all-action modeling might be more efficient because it can exploit shared concepts of action quality among actions. Through experimentation, they confirmed that their approach is better able to use data, is more generalizable and adaptable.

Parmar *et al.* [29] revisited the definition of AQA, and noted that it was in fact a function of functions – what action was performed, and how well that action was performed. Instead of op-

timizing the single AQA function, like all the exiting approaches, they proposed to optimize all the three functions simultaneously. To execute and their evaluate their approach, they release a novel multitask AQA dataset. They showed that multitask learning (MTL) approach outperforms single-task learning approach. In addition, in their work AQA-oriented representations were learnt in end-to-end fashion.

Longer action sequences, like in case of Figure Skating, can contain a lot of irrelevant information in between sparsely distributed program elements. To mitigate with this, Xu *et al.* [42] proposed to use self-attentive and multiscale skip convolutional lstm to aggregate information from individual clips. They also extend the existing MIT-Figure Skating dataset to 500 samples. Their method achieves best performance on the assessment of Figure Skating samples.

On physiotherapy side, Parmar *et al.* [27] compile a new physiotherapy dataset. Dataset contains samples where the human subjects are doing exercise in an ideal way and samples where they are deliberately making mistakes in execution of exercise. Data consists of positions of joints as captured a Microsoft Kinect sensor. In their pilot study, they compare four classifiers to see how accurately can they identify erroneous samples. For a related domain of squatting exercise, Ogata *et al.* [24] release a new squatting exercise dataset. In their work, they develop approaches which are aimed at identifying the type of error made by human subjects.

Movement Quality Assessment : A portion of works is focused on movement analysis, which we discuss in the following. Paiement *et al.* [25] develop an approach to assess the quality of human motion from skeleton data captured using Microsoft Kinect sensor. Precisely, they

validate their method on assessment of gait on stairs. Skeleton data is noisy and with higher dimensionality. To handle noise and reduce the dimensionality of data, a robust non-linear manifold learning technique is used. A statistical model of gait is built from movement of healthy subjects. A test sample is assessed by matching with the healthy model on a frame-by-frame basis following Markovian assumptions. Work by Paiement *et al.* was extended by Tao *et al.* [37] and additionally tested for analyzing the quality sit-stand motion and gait on flat surface.

Given heat-maps and limb-maps of human subjects, Sardari *et al.* [33] train a CNN regression network to transform those maps into view-invariant representations using a manifold. They also compile a multiview, multimodal dataset with application in physiotherapy.

We summarize works related to AQA in Table 2.1, and works focused on movement analysis in Table 2.2.

Paper	**Dataset**	**Application**	**Approach**	**Key idea**	**Loss function**	**Result (best approach) (SR in %)**
[31]	MIT-Dive	Olympic Judging	STIP; ConvISA; Pose features + DCT/DFT + SVR	AQA; highlights; feedback	SVR; Ridge regression	41
	MIT-FS					45
[28]	UNLV-Dive	Olympic Judging	C3D + SVR; C3D + LSTM + fc; C3D + LSTM +SVR	Visual cues	SVR; L2	79.02

Table 2.1 continued from previous page

Paper	**Dataset**	**Application**	**Approach**	**Key idea**	**Loss function**	**Result**
	UNLV-FS					53
	UNLV-Gym-vault					68.24
[26]	AQA-7-Dive	Olympic Judging	C3D + LSTM + fc (consolidated dataset)	AQA across multiple actions	L2	61.77
	AQA-7-GV					67.46
	AQA-7-BigSki					49.55

Table 2.1 continued from previous page

Paper	**Dataset**	**Application**	**Approach**	**Key idea**	**Loss function**	**Result**
	AQA-7-BigSnow					36.48
	AQA-7-Sync-Dive-3m					84.1
	AQA-7-Sync-Dive-10m					73.43
[41]	UNLV-Dive	Olympic Judging	S3D	Segmented approach	L2	86

Table 2.1 continued from previous page

Paper	**Dataset**	**Application**	**Approach**	**Key idea**	**Loss function**	**Result**
[20]	UNLV-Dive	Olympic Judging	3D	Segmented approach, E2E	L2 + Ranking Loss	80.09
	UNLV-GV					70.28
	UNLV-FS					57.53
[21]	MIT-Dive	Olympic judging	3D features of key fragments	KFS	L2+RankingLoss	78
	UNLV-Dive					84

Table 2.1 continued from previous page

Paper	**Dataset**	**Application**	**Approach**	**Key idea**	**Loss function**	**Result**
	UNLV-GV					70
[29]	UNLV-Dive	Olympic judging	C3D-AVG features	MTL approach	L2+L1+Detailed action recognition loss + commentary loss	88.08
			MSCADC			80.6
	MTL-AQA		C3D-AVG features			90.44
			MSCADC			86.12

Table 2.1 continued from previous page

Paper	**Dataset**	**Application**	**Approach**	**Key idea**	**Loss function**	**Result**
[42]	MIT-Skate	Olympic judging	C3D-LSTM	self-attentive + multiscale skip convlstm	L2	59
	Fis-V					78
[24]	Squatting dataset	Squatting	skeleton		cross-entropy	
[27]	exercise dataset	physiotherapy	skeleton		classification	94.68

Table 2.1: Summary of AQA works.

Work	Dataset	Area of Application	Approach/features
[25]	SPHERE-StairCase	stair climbing gait analysis	Skeleton jointsfrom Kinect. Non-linear manifold learning followed by modeling normal movement by healthy subjects.
[37]	SPHERE-SitStand; SPHERE-Walking	stair climbing gait analysis; flat surface gait analysis; sit-stand	Skeleton jointsfrom Kinect. Non-linear manifold learning followed by modeling normal movement by healthy subjects.
[33]	SMAD	movement analysis in physiotherapy	pose projected onto a view-invariant manifold

Table 2.2: Summary of movement analysis works.

Skills Assessment

Zia *et al.* [46], transform HoG-HoF descriptors into frequency domain, which forms spatiotemporal interest points (STIP's) for their approach. Finally, a classifier is learnt that can correctly classify the STIP's into three surgical skills level – *novice*, *intermediate*, and *expert*.

Doughty *et al.* [7], also develop a CNN-based approach to assess surgery skills, where the CNN is trained using a pairwise ranking loss. Additionally, they also show that their approach

works for skills determination in general scenarios like dough rolling, chopstick using, and drawing.

Li *et al.* [22] proposed to incorporate attention mechanism to attend to salient areas footage. By incorporating attention, their approach was able to push forward the state-of-the-art not only dataset intorduced by Doughty *et al.* [7], but also on their own new dataset, comprising of samples of infants grasping and manipulating objects.

In addition to incorporating attention, Doughty *et al.* [8] proposed to use a modified loss function consisting of rank-aware ranking and disparity loss. They also introduced new actions to skills determination datasets.

Skills assessment has been applied to sports as well, for *e.g.*, Ilg *et al.* [12], propose an approach to estimate skills in Karate Kata based on hierarchical spatiotemporal correspondences. They introduce and use their own dataset which contains vicon data of joint position while human subjects are performing Karate Kata. Bertasius *et al.* [4] present an approach to determine basketball skills by analyzing egocentric videos. Atomic basketball events are first detected that are then passed through a Gaussian mixtures, which gives features that would be indicative of player's skills. Finally, these features are mapped to skills level. Since they have their dataset annotated by a single basketaball coach, annotations are likely to be subjective.

Skills assessments works are summarized in Table 2.3.

Work	**Dataset**	**Appli-cation**	**Approach**	**Key idea**	**Loss function**	**Result**
[7]	jigsaws	skills determination	2d-3frames		pairwise ranking	70.2
	dough rolling					79.4
	chopstick					83.2
	drawing					71.5
[22]	infant grasp	skills determination	RGB image + optical flow: CNN	feature encoding + attention pooling + temporal aggre-gation	pairwise ranking	86.1
	jigsaws					73.1
	dough rolling					82.7
	chopstick					85.5

Table 2.3 continued from previous page

Work	Dataset	Appli-cation	Approach	Key idea	Loss function	Result
	drawing					85.3
[8]	EPIC-Skills	skills determination	I3D	attention + loss	ranking + disparity + rank-aware	80.3
	BEST					81.2
[46]	own (un-named)	skills determination	STIP + DCT/DFT + nearesr neighbor		various	100

Table 2.3 continued from previous page

Work	**Dataset**	**Appli-cation**	**Approach**	**Key idea**	**Loss function**	**Result**
[4]	first-person basket-ball	skills determination	CNN + gaussian mixtures		hinge loss	79.3
[12]	karate kata	skills estimation	vicon data		distance func-tion	

Table 2.3: Summary of Skills Assessment works.

CHAPTER 3

LEARNING TO SCORE OLYMPIC EVENTS

Introduction

As mentioned in Chapter 2, prior works on AQA had used human body pose features to represent action quality. Although pose features are expressive, they neglect important visual cues like splash during entry into the water; and estimating pose correctly is especially difficult in sports actions, where the athletes undergo convoluted pose transformations. As a note, human judges do take these visual cues into consideration. So, in this chapter, we try to seek answers to following questions:

1. How can we bypass poor pose estimation problem?

2. How can we incorporate available visual cues?

In addition, we compile a new bigger AQA dataset to address the data shortage in AQA.

Convolutional neural networks (CNN's) ([10]; [18]) are the state-of-the-art approaches for almost all the computer vision tasks such as object recognition, action recognition, etc. In traditional feature designing approaches (as opposed to CNN's) prior to deep learning era (2012 onward) features/representations were hand engineered, which were geared toward the task at hand. In CNN's, a hierarchy of representations are learnt/optimized automatically using backpropogation algorithm [32]. Following this success, we propose to use CNN's fea-

tures/representations, instead of pose features, for the task of AQA. CNN's come in many flavors, like 2D CNN's, 3D CNN's, graph CNN's, etc. 1D CNN's are suitable for a single dimensional signals, likewise, 2D CNN's are suitable for 2D-dimensional data like images, and 3D CNN's are suitable for 3D data like videos. Actions may be represented in using a single image, but such a representation would be ambiguous, for *e.g.*, if capture an image of a person halfway getting up from a chair, then without any context or past or future information, it would be difficult to say whether the person was halfway getting up from the chair, halfway sitting into the chair. So, while a single image might be sufficient to represent actions for action recognition task, videos are a better choice to represent actions when measuring its quality. Moreover, videos have experimentally been shown to yield better results on the task of action recognition. So, in particular we propose to use 3D CNN's (C3D [38]).

Approach

In computer vision, we represent input data (in our case videos of actions performed by athletes) using features of much lower dimensions which capture the task-oriented gist of the input. In our case, end task is action quality assessment, so features would be capturing that. CNNs can be thought of as a large no. of filters placed in cascaded manner, through which when input data is passed, useful properties are retained and becomes the part of output feature/representation. 3D CNNs are memory and computationally intensive, which makes them suitable to process only small video clips (16 frames). Current 3D CNNs are not suitable for processing longer videos like ours, which are 100 frames long. Unlike action recognition, which can be performed by seeing as little evidence as a single video frame, in order to assess

the quality of an action in a meaningful way, full action sequence needs to be processed. A way to address to longer videos is making their spatial dimensions much smaller. This might work well for action recognition, where finer details are not of much importance, but finer details, for example, whether athletes legs are bent or straight, are important in our case, so this approach of spatial dimensions reduction might not be well-suited for our case. Therefore, instead, we propose to breakdown our long videos (100 frames) into smaller clips (16 frames), compute the features for individual clips, and then combine these clip-level features in order to obtain video-level feature/representation.

Computing spatiotemporal features

Neural networks, Convolutional or otherwise, need to be trained a dataset in order to optimize its tunable parameters (weights and biases) such that it makes correct prediction/estimation on the end-task. 3D CNNs have larger number of parameters than a 2D CNN, which means it will require larger dataset to train effectively. AQA datasets, unlike action recognition datasets are smaller, so it is not wise to train a 3D CNN on our small AQA datasets.

CNNs consist of Convolutional layers, followed by linear layers/fully-connected layers. The idea behind this kind of design is that Convolutional layers learn to encode input data (images/videos) into much less dimensional codes (features), and then the linear layers, which sit on top of these Convolutional layers, learn to decode these codes into correct task-oriented outputs. Therefore, while last couple of fully-connected layers are heavily taskoriented, the Convolutional layers are generalizable, and can be reused for other tasks. Therefore, CNNs, unlike hand engineered features, learn features which can work well across tasks. This means

that we can effectively train a CNN using a larger dataset on all together a different task, and then chop-off the task-oriented fully-connected layers and reuse the Convolutional layers. This is a very well-known process called pretraining. We can take the pretrained Convolutional layers as they are and simply learn a couple of more layers oriented for our task on top of them. We observe that action recognition is closely related task to ours AQA task, and therefore hypothesize that action recognition features might transfer well on to our task.

Frameworks

We propose the following three frameworks to perform AQA using spatiotemporal representations learnt by CNNs: C3D-SVR, C3D-LSTM, C3D-LSTM-SVR. Our proposed frameworks differ from each other in the following ways:

1. in the way they accumulate/aggregate/combine evidence from individual clips (clip-level features) to form video-level representation

2. in the regression schemes, they use to map video-level features to AQA scores.

Aggregation schemes

1. Averaging: On the task of action recognition, it was observed that element-wise averaging activations coming out of one of the fully-connected layers, is a good way to combine clip-level features to obtain video-level representation. So, we follow this scheme for our AQA task as well.

2. Long Short Term Memory [11] Recurrent neural networks (RNN's) are used for modeling

time series data. Clip-levels features can be viewed as time series data. Final score (or equivalently, video-level representation) is a function of what (and *how well*) happened in each of the clips because in RNN, by design, the current output is a function of current hidden state and the previous hidden state. RNN's are a way to model these long-term dependencies. Vanilla RNN's suffer from vanishing gradients. Long Short Term Memory (LSTM's) are a type of RNN cell which was developed to mitigate vanishing gradients problem. RNN/LSTM also benefit from sharing weights across all time steps. Therefore, we propose to use LSTM to aggregate evidence from all the clips to make final decision.

Regression models

Once we have the video-level representations, the next step is to learn to map those representations to actual AQA scores (as standardized by FINA [2] and given by the judges). Mapping the action representation to actual scores is a regression task. We consider the following two regression models.

1. Fully-connected layers: In this scheme, we use linear layers to decode the video-level representation to a single number, AQA score.

2. Support Vector Regressor (SVR) [9]: SVM/SVR are well known class of maximum margin classifiers/regressors. Soft-margin SVR might work better than fully-connected layers, so we also consider experimenting with SVR.

Combinations of aggregation and regression schemes give us three frameworks as in Table 3.1. Full pipelines are illustrated in Fig. 3.1.

Framework	Aggregation scheme	Regression scheme
C3D-SVR	Averaging	SVR
C3D-LSTM	LSTM	fc
C3D-LSTM-SVR	LSTM	SVR

Table 3.1: Frameworks.

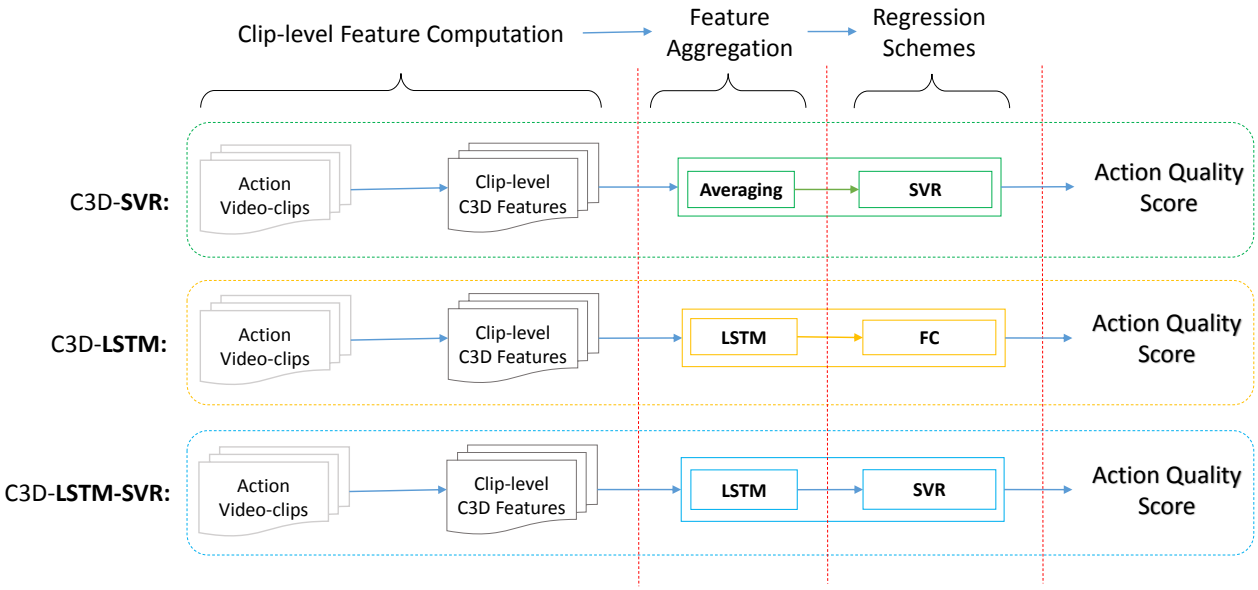


Figure 3.1: Frameworks.

Datasets

We consider Olympic events to train and evaluate our frameworks. Wnuk *et al.* [40] and then Pirsiavash *et al.* [31] first proposed to consider using Olympic events for developing and evaluating AQA approaches. Using Olympic events have following benefits:

1. Olympic events have well-defined objective scoring criteria.

2. Scoring is carried out by expert human judges, who had to undergo extensive training and

licensed by professional organizations, so their judgment is very reliable, and moreover, final score is effectively an average of multiple judges (generally, 5 to 7 judges).

3. Recorded video footages of Olympic events are nowadays available in abundance, thanks to video sharing platforms like YouTube; and these footages naturally come with judges' scores. These allows for creating larger datasets.

Pirsiavash *et al.* [31], introduced Diving and Figure skating datasets with 159 and 150 samples, respectively. We first of all extend the Diving and Figure skating datasets to 370 and 170 samples, respectively, and then introduce a new dataset, Gymnastic Vault. We further discuss these Olympic events in the following:

Diving: There are different types of diving events differentiated on basis of board height (1m, 3m, 10m), board type (platform, springboard), individual/synchronous events. In our dataset, we consider 10m platform men's individual event that had taken place London Olympics 2012. We compiled dataset samples from quarter finals, semifinals, and finals. All the samples have been video recorded from the side view, with almost no view variation. View variation can play an important role in a computer vision system's performance. Final score, in case of diving, is a product of dive execution quality score, and dive difficulty score. Average length of a diving dataset sample is around 100 frames long, recorded at a rate of 25.5 frames/second.

Figure skating: There are two types of figure skating events: short program and full program. We consider short program. Final score is a sum of a base technical score and presentation score. Each sample is around 4500 frames long (around 2.5 minutes). Camera position and

camera angles are switched throughout the event, so there's a continuous change of camera positions and angles.

Gymnastic vault: We compile a gymnastic vault dataset with 176 samples. Average length of samples is around 75 frames. View remains constant during a sample (unlike figure skating), there's a large view variation across samples (unlike diving). We collected our dataset from 13 events, so there's a lot variation in the background as well.

Experiments

Objective function: To train LSTM aggregation and fully-connected regression layers we follow minibatch stochastic gradient descent, and backpropagation algorithms, where we consider the Euclidean distance between predicted AQA scores and groundtruth AQA scores as the objective function, Eq. 3.1, to be minimized.

$$\mathcal{L}_{AQA} = |Scores_{pred} - Scores_{GT}|^2 \tag{3.1}$$

Implementation details: We implement the CNN's in Caffe framework. We consider original C3D network, pretrained on Sports-1M action recognition dataset. In addition to that, we also consider a smaller C3D architecture, which has reduced number of convolutional layers, and pretrained on UCF101 action recognition dataset. For LSTM based frameworks, we temporally normalize all the action sequences to 103 frames – we drop frames if a sequence has more than 103 frames, or if there are less than 103 frames we insert zero frames in the starting of the sequence to make a total of 103 frames. For optimizing LSTM and fully-connected layers,

we use ADAM as our solver. We do apply spatial and temporal augmentation while training.

Metrics: We use Spearman's rank correlation, ρ, as the performance metric for our frameworks.

We evaluate our frameworks and also compare it with the state-of-the-art approach by Pirsiavash *et al.* [31], to see if hypothesis was true that performance on AQA task can be improved by taking into account visual cues. Results for Diving, Gymnastic Vault and Figure Skating actions are shown in Tables 3.2, 3.3, and 3.4, respectively. For figure skating, where action sequences are very long, we only consider C3D-SVR.

Method	**Spearman's rank correlation**
Pose + DCT [31]	53.00
Ours C3D-SVR	**79.02**
Ours C3D-LSTM	68.66
Ours C3D-LSTM-SVR	70.06

Table 3.2: Performance comparison on Diving action.

Method	Spearman's rank correlation
Pose + DCT [31]	10.00
Ours C3D-SVR	67.57
Ours C3D-LSTM	67.66
Ours C3D-LSTM-SVR	**72.23**

Table 3.3: Performance comparison on Gymnastic Vault action.

Method	Spearman's rank correlation
Pose + DCT [31]	35.00
ConvISA [19]	45.00
Ours C3D-SVR	**53.00**

Table 3.4: Performance comparison on Figure Skating action.

Discussion of results

Diving action: We can see that taking visual cues into account improves the performance over just pose based method, which supports our hypothesis. Secondly, comparing our frameworks with each other, we see that C3D-SVR works better than LSTM based frameworks. Samples in Diving dataset do not have much view variation. So, learning SVR, which is a margin maximizing algorithm is able to model pretty good in feature space. LSTM-based frameworks can do more complex modeling than SVR. But since there's no view variation, a simple model like C3D-SVR performs better, and adding complexity might not be helping.

Gymnastic vault: Pose + DCT uses a pose extraction method that needs to be optimized for each action. We have used their publicly released code, where the pose extraction is optimized for diving and figure skating. Pose extraction not being optimized for Gymnastic Vault might be a reason for Pose + DCT yielding very poor results on Gymnastic Vault. Other reason for its poor performance as compared to diving action, might be the large view variation. Comparing our results, we see that for Gymnastic Vault, where view variation is large, LSTM-based approaches outperform C3D-SVR.

Figure Skating: For figure skating as well, considering visual cues helps better measure its quality. Pose estimation seems to be affected negatively due to view variation. Visual feature based methods, ConvISA and ours work better than pose based approach.

Conclusion

In this work, we hypothesized that pose-based approaches do not take into account visual cues available from the raw video footage, which might be hurting their performance on the AQA task. So, to mitigate that, we proposed three frameworks, which differed in the way they aggregated clip-level features, and the regression module. Our frameworks, which take into visual cues outperform pose-based methods by a good margin.

CHAPTER 4

ACTION QUALITY ASSESSMENT ACROSS MULTIPLE ACTIONS

Introduction

In last chapter we transitioned the AQA function from a classifier to a regressor. In this chapter, we continue to treat AQA function as a regression function, but propose to learn it in a more efficient way.

Current AQA approaches, including the one discussed in the previous chapter, train an action-specific models, i.e., a separate model is trained for each action. In this chapter, we try to seek answers to following questions:

1. Are there common/shared action quality elements among different actions?
2. If so, would it beneficial to train/pretrain a single/shared model across various actions?
3. For zero-shot AQA, will a model pretrained on multiple actions perform than a model pretrained on a single action?

Since current approaches learn action-specific models, they don't exploit the fact that there are (some) shared, common action quality elements/concepts among different actions. Our hypothesis is that by exploiting these common action quality elements, we can improve the performance on the task of AQA. First of all, we introduce a new multi-action AQA datasets,

followed by a discussion of common action quality elements. Then, we discuss our approach and test our hypothesis in the experimental section.

AQA-7 Dataset

There's a dearth of AQA datasets – not only in terms of sheer number of datapoints, but also the variety of actions covered. To mitigate with this, we introduce a novel AQA dataset, where we not only increase the datapoints, but also include more actions. Our newly compiled dataset has seven actions from sports, which have clear, objective criteria. We discuss these actions in the following.

Diving: We consider both individual and synchronous diving events, and consider both 3m springboard and 10m platform mens and womens events in compiling our dataset. Individual diving was forst introduced in Olympics in early 1990's, followed by synchronous diving in early 2000's. In individual diving events, only how well the diver performed is evaluated, while in synchronous events, where two divers perform the same dive, synchronization of maneuvers between both the divers is also given importance.

Gymnastic vault: As discussed in Chapter 3.

Big Air Skiing and Snowboarding: There are many types of skiing and snowboarding events like slalom, half-pipe, etc. We choose to consider Big Air events (BigSki and BigSnow). Before, Winter Olympics 2018, X-Games was the premier venue for these events, so we use X-Games footages to collect dataset samples. There's a large view variation among events and

even within individual dataset samples as numerous moving camera are used to capture the footages. Score takes into account four components: difficulty, execution, amplitude/height achieved by the athlete, progression and landing. BigSki and BigSnow use a rather more complicated formula to calculate final score, since athlete is awarded if they push forward the sport itself by attempting tricks that no one else has done.

Trampoline: Trampoline was first included in Olympics in the year 2000. Final score is based judging the following: difficulty level, execution quality, and time of flight (how long the athlete was in air). Unlike other actions that we have covered so far, where a sample consisted of only action phase, trampoline samples consist of about 10 action phases. In each action phase, the athlete jumps off the trampoline; and does tricks while going up and coming down. Each trampoline sample is way longer (around 650 seconds) than previously discussed actions because of the multiple action phases.

Please refer to Fig. 4.1 for a preview of our dataset. We present the action-wise details/characteristics of AQA-7 in Table 4.1.

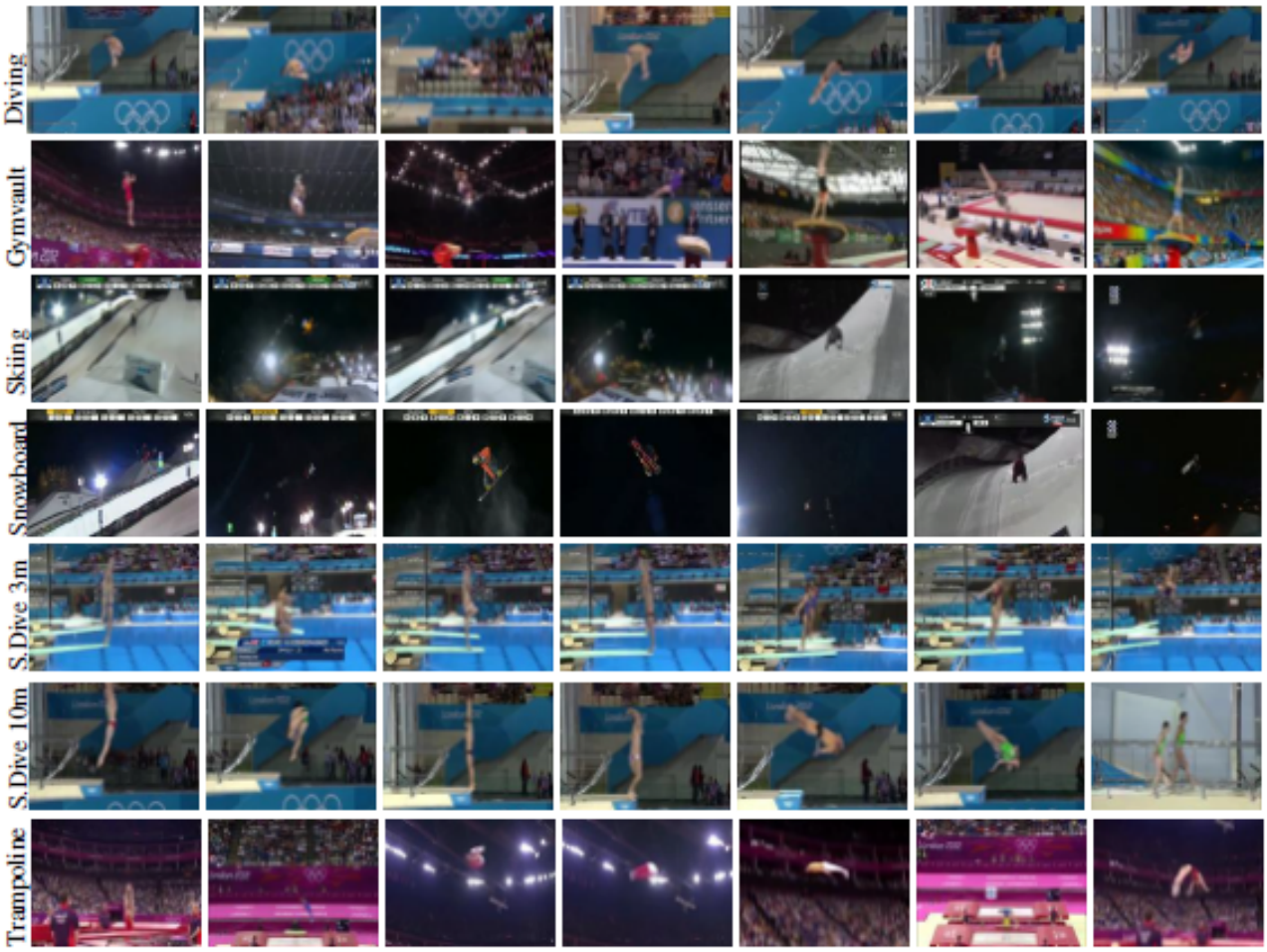


Figure 4.1: Preview of AQA-7 dataset.

Sport	**Avg. Seq. Len.**	**# Samples**	**Score Range**	**# Participants**	**View Variation**
Single Diving 10m platform	97	370	21.60 - 102.60	1	negligible
Gymnastic vault	87	176	12.30 - 16.87	1	large
Big Air Skiing	132	175	8 - 50	1	large
Big Air Snowboarding	122	206	8 - 50	1	large
Sync. Diving 3m springboard	156	88	46.20 - 104.88	2	negligible
Sync. Diving 10m platform	105	91	49.80 - 99.36	2	negligible
Trampoline	634	83	6.72 - 62.99	1	small

Table 4.1: Characteristics of AQA-7 dataset.

Common Action Quality Elements

The sport actions present in our AQA-7 dataset have action elements that affect the AQA scores in the same way. For *e.g.*, bad landing negatively affects the AQA score in gymnastic vault, BigSki, and BigSnow, while having perfectly straight legs helps improve the AQA score in case of gymnastic vault and diving events. More such examples are in Fig. 4.2. Not all the elements are shared among all the actions. There are action-specific elements as well, which are not shared.

Now, we explain the reason behind the occurrence of common action quality elements. No matter whether you are going to enter water (diving) or land on mat/snow (gymnastic vault, BigSki, BigSnow), having to complete higher number of twists or somersaults or spins (difficulty aspect) in the limited time from take-off to entry into water/landing, while keeping legs straight and body in a tight tuck/pike position (related to execution quality) is harder to achieve, and therefore, worthy of more points from judges, or equivalently, higher action quality.

Having observed that certain action quality elements are shared, our hypothesis is that knowledge of what aspects to assign more points in one action can be transferred to other actions.

Approach

In order to exploit common action quality elements, we need to use a model that can share the common action quality elements. We propose to use C3D-LSTM framework because of the following reason. LSTMs are used for processing time series data, like in our case; and they also have and learn their own internal representation of the data (hidden state). In nutshell, hidden

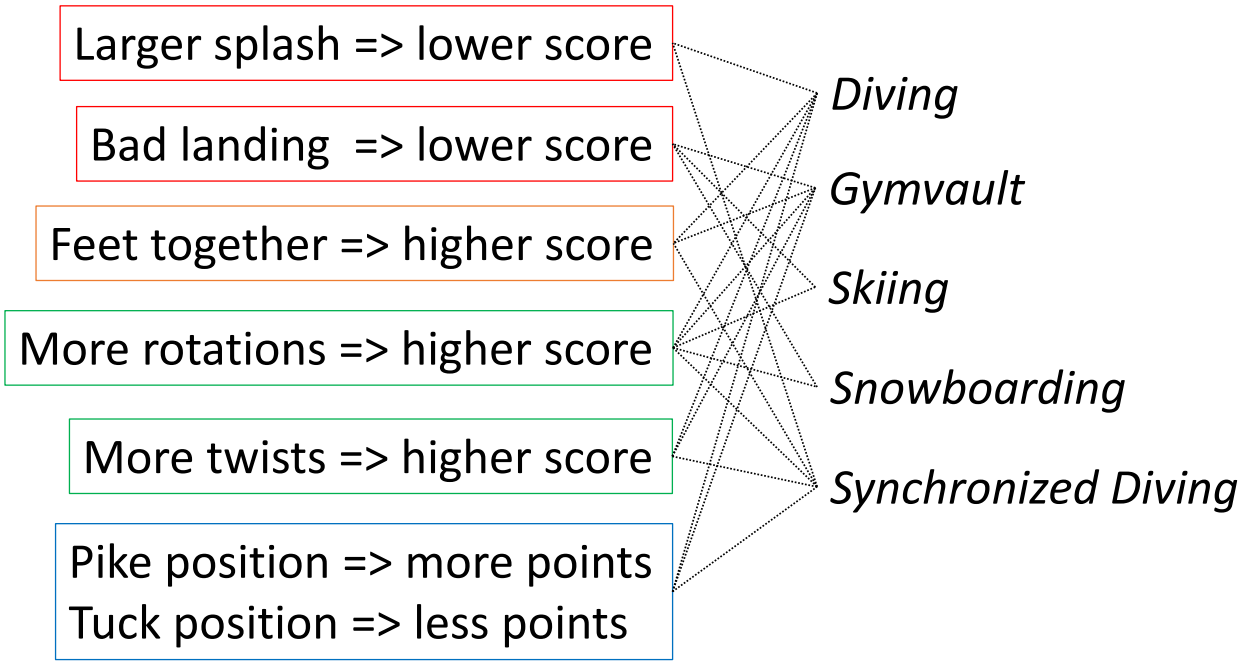


Figure 4.2: Illustration of common action quality elements.

states are a function of previous hidden states and the current input. During training, weights inside LSTM are learnt. The LSTM output of the last step (256-dimensional, in our case) is then mapped to a final score using a FC layer. Our approach utilizes the training data more efficiently to learn weights in LSTM and FC layer, by exploiting the common action elements by jointly training across all actions. We believe training internal LSTM weights and the fully connected output regression layer better represents the underlying structure of the sports actions and allows sharing of common elements through more effective use of limited data. Each action, individually, has very less number of datapoints, but when combined together, we can get larger number of datapoints, which essentially means we have more data per action element to learn from.

Experiments

We design and carry out experiments with the following motives:

1. As a sanity check to see if it is possible to a single consistent across multiple actions, and if so, to compare that all-action against action-specific models

2. Zeroshot AQA - to evaluate how well an all-action model can quantify the quality of unseen action classes

3. Adaptation/finetuning to novel/unseen action classes using smaller datasets

Performance metric: Spearman's rank correlation is used as the performance metric. For aggregated/averaged results, we present average Spearman's rank correlation, computed from individual action-wise correlations using Fisher's z-value as described in [36].

Data preparation: Since actions have different score ranges, we normalized the scores by dividing the raw scores with the standard deviation of the corresponding action.
All the sequences are normalized to 103 frames. We employ temporal data augmentation.

Implementation: C3D-LSTM framework is implemented in Caffe [15] on Titan-X GPU. We use C3D pretrained on UCF101 action recognition dataset. During the experiments, we freeze the C3D network, and only learn the LSTM parameters. For which, ADAM solver [17] is used with an initial learning rate of 0.001 and annealed by a factor of 2 after every 3,000 iteration. Optimization is carried on for a total of 20,000 iterations, with a batch size of 15 samples. LSTM layer is initialized with Gaussian noise of standard devaition of 0.1.

All-action vs. Action-specific modeling

In this section, we compare our all-action model (model trained on consolidated dataset from multiple actions) with two action-specific models: C3D-SVR, and C3D-LSTM, which are state-of-the-art approaches. The baseline for our all-action model is C3D-LSTM since both have same aggregation scheme (LSTM) and regression model (fully-connected layer).

In comparison to action-specific C3D-LSTM model, all-action model performs better at measuring the quality of five out of six action classes - all action classes, except Snowboarding class for which performance dropped by 0.14 (in terms of Spearman's rank correlation). Without making any changes to the network design, on an average, all-action model outperforms by 3% by leveraging data samples from multiple actions.

It shall be noted that our strategy to train a single model using datapoints from all actions is complementary to the existing approaches, which can help improve their performance.

Zero-shot AQA

Whether the learned concepts of quality of action can translate across actions is an open question in the area of AQA. We try to seek answer to this question in this section. The knowledge of how to measure the quality of multiple actions is likely to help in measuring the quality of other, unseen actions, if concepts of quality of actions are shared among actions.

The previous, although it showed that all-action C3D-LSTM model performed better than action-specific model, it necessarily did not support the idea that learning to assess the quality of one action helped with the assessment of others. To that end, we devise another experiment, where we train a model on datapoints from five actions, and test it to measure the quality of

samples from the sixth, unseen action class.

We need baselines for comparison to put the result of experiment in perspective. We consider the following baselines.

Random-initialization vs. Multi-action pretraining

For the first baseline, we initialize the parameters of LSTM and fully-connected layers with random Gaussian noise; C3D weights remain same. Comparison with this baseline may not seem very fair, but in an interesting work by Jarrett *et al.* [14], it was shown that a hierarchy of randomly initialized, untrained convolution filters perform almost as good as learned filter weights. Also, if action quality concepts were not shared among actions, then our multi-action model should have performance similar to randomly initialized model. However, if the multi-action model outperforms randomly-initialized one, then it is indicative of shared action quality elements, and that there is a utility in learning an all-action model.

From the results, we can see that randomly-initialized (RI) model performs with nearly no reliability – all teh correlation values are sitting close to zero. All-action model in comparison seems to be working better, indicating the existence of shared / common action quality elements. However, for Gymnastic vault and Skiing, all-action model does not seem to work better. We further explore these cases in Section, and indeed find that multi-action training provides a good initialization.

Single-action vs. Multi-action transfer To further examine the idea of knowledge transfer, we compare the performance after transferring from a single action to an unseen action versus the performance after transferring from a group of five actions. The main hypothesis behind this

experiment is that more action quality elements, then transferring from a single action, would be shared if we transfer concepts from multiple actions to unseen action. If this hypothesis were correct then higher performance boost should be there when transferring from multiple actions then a single action.

The results are shown in Table. As expected, for single action transfers, model works best when train and test class are same. But some non-intuitive relationships also emerge; for *e.g.*, a model trained on BigSki works well on Diving action.

More importantly, we can see that, on an average, multi-action transfer outperforms any single action transfers by a good margin, which supports our hypothesis. This indicates that as we increase our action bank (include more actions), the likelihood of sharing action quality elements with an unseen action increases.

Finetuning to novel action classes

In AQA, datasets for many actions are many a times very small. So, in this experiment, we use finetuning to adjust multi-action-pretrained and randomly-initialized to a new, unseen action class using very few training samples. And then compare multi-action-pretrained vs. randomly-initialized models. We set the hyperparameters proportional to training set size.

General trend that we observe is the multi-action pretrained is better at adapting to the novel action classes and needs very few training iterations to do so. Gymnastic Vault and BigSki which seemed to have poor initialization from all-action pretraining, adjust quickly and with good performances.

Conclusion

We demonstrate that AQA can benefit from knowledge transfer through sharing action quality elements among actions from similar domains. We compiled the largest AQA dataset yet from seven actions. Experiments confirm that all-action modeling can: i) make better use of data, ii) provide better generalizability, iii) better adapt to new action classes. We tried to keep resetting of hyperparameters to minimum. This kind of approach can be exploited in scenarios other than sports, like, for *e.g.*, in surgical skills where skill concepts can be shared among knot-tying, needle-passing, and suturing.

	Diving	**Gymvault**	**Skiing**	**Snowb-oarding**	**Sync. Dive 3m**	**Sync. Dive 10m**	**Avg. Corr.**
Pose+DCT [31]	0.5300	-	-	-	-	-	-
Single-action C3D-SVR[28]	**0.7902**	**0.6824**	**0.5209**	0.4006	0.5937	**0.9120**	**0.6937**
Single-action C3D-LSTM[28]	0.6047	0.5636	0.4593	**0.5029**	0.7912	0.6927	0.6165
Ours All-action C3D-LSTM	0.6177	0.6746	0.4955	0.3648	**0.8410**	0.7343	0.6478

Table 4.2: All-Action vs. Single-Action models. Performance evaluation of single-action and all-action models in terms of action-wise and average Spearman's rank correlation (higher is better). First two frameworks simply average features to aggregate them and use SVR as the regression module. The bottom two frameworks use LSTM to aggregate features and use a fully-connected layer as the regression module. Our approach can be directly compared with single-action C3D-LSTM [28], since both have the same architecture.

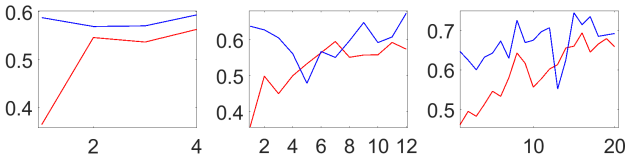
	Unseen test action class						
	Diving	**Gymvault**	**Skiing**	**Snowboard**	**Sync. Dive 3m**	**Sync. Dive 10m**	**Avg. Corr**
Random Wts./Ini.	0.0590	0.0280	-0.0602	-0.0703	-0.0146	-0.0729	-0.0218
Diving	**0.6997**	-0.0162	0.0425	0.0172	0.2337	0.0221	0.0599
Gymvault	0.0906	**0.8472**	0.0517	0.0418	-0.1642	-0.3200	-0.0600
Skiing	0.2653	-0.1856	**0.6711**	0.1807	0.1195	0.2858	0.1331
Snowboard	0.2115	-0.2154	0.3314	**0.6294**	0.0945	0.1818	0.1208
Sync. Dive 3m	0.1500	-0.0066	-0.0494	-0.1102	**0.8084**	0.0428	0.0053
Sync. Dive 10m	0.0767	-0.1842	0.0679	0.0360	0.4374	**0.7397**	0.0868
Multi-action	0.2258	0.0538	0.0139	0.2259	0.3517	0.3512	**0.2037**

Table 4.3: Zero-shot AQA. Performance comparison of randomly-initialized model, single-action models (for , first row shows the results of training on diving action measuring the quality of the remaining (unseen) action classes), and multi-action model (all-action model trained on five action classes) on unseen action classes. In multi-action class, the model is trained on five action classes and tested on the remaining action class (column-wise). In single-action model rows, diagonal entries show results of training and testing on the same action. Avg. Corr. shows the result of average (using Fisher's z-score) correlation across all columns.

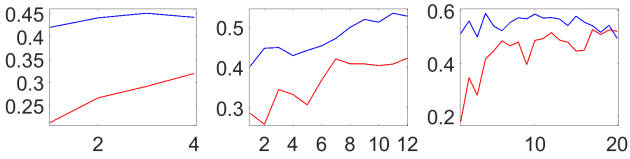
Test action	**Diving**			**Gymvault**			**Skiing**		
# samples	**25**	**75**	**125**	**25**	**75**	**125**	**25**	**75**	**125**
RI	0.5633	0.5952	0.6935	0.3197	0.4231	0.5278	0.0955	0.5050	0.5862
AA	**0.5937**	**0.6742**	**0.7443**	**0.4509**	**0.5350**	**0.5894**	**0.1279**	**0.5778**	**0.5991**

Snowboard			**Sync. Dive 3m**			**Sync. Dive 10m**		
25	**75**	**125**	**15**	**25**	**35**	**15**	**25**	**35**
0.1813	**0.4507**	**0.4751**	0.2659	0.3382	0.4268	0.3511	0.4913	0.6305
0.1978	0.3347	0.4437	**0.5235**	**0.5980**	**0.7429**	**0.4500**	**0.7900**	**0.8123**

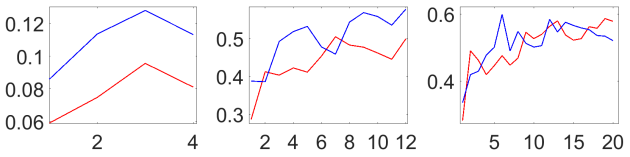
Table 4.4: Finetuning from scratch vs. finetuning from pre-trained multi-action model. Experimental results (Spearman's rank correlation) of finetuning a randomly-initialized (RI) model and an all-action (AA) model pre-trained on five action classes. The numbers represent the best results from all the iterations.



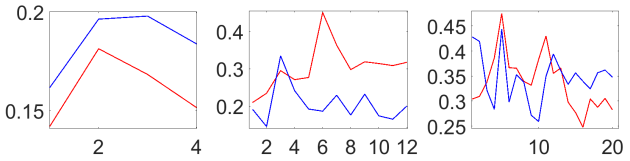
(a) Diving - 25 (b) Diving - 75 (c) Diving - 125



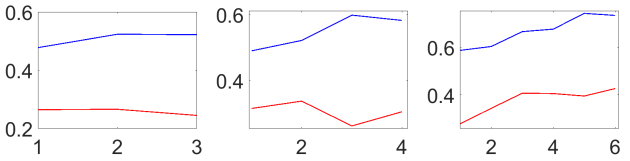
(d) Gymv - 25 (e) Gymv - 75 (f) Gymv - 125



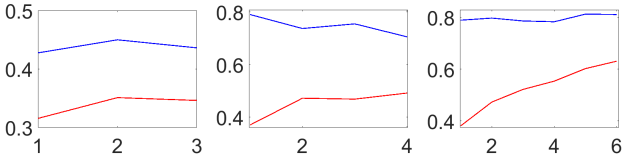
(g) Ski - 25 (h) Ski - 75 (i) Ski - 125



(j) Snowb - 25 (k) Snowb - 75 (l) Snowb - 125



(m) S.D 3m - 15 (n) S.D 3m - 25 (o) S.D 3m - 35



(p) S.D 10m - 15 (q) S.D 10m - 25 (r) S.D 10m - 35

Figure 4.3: Finetuning from scratch vs. finetuning from pre-trained multi-action model. Plot of Spearman's rank correlation against every hundred iterations for different number of training samples. Blue and red curves represent multi-action and randomly initialized models, respectively. The gap in the initial iterations suggest that good initialization of LSTM weights was achieved by training on multiple actions. In most of the cases, multi-action model has better performance than randomly initialized model on test samples throughout all the iterations.

CHAPTER 5

MULTITASK LEARNING APPROACH TO ACTION QUALITY ASSESSMENT

Introduction

Existing works in skills assessment and AQA use a single label – final score – to train some kind of machine learning model. A single score may not be sufficient to characterize a complex action, for *e.g.*, a diving routine. Due to this poor characterization, the performance of these might be limited. In this work, we revisited the definition of AQA, and observed that AQA is in fact a function of what action performed and how well that action was performed. So, we pose the following question:

1. Can learning to describe an action in detail help improve the performance on AQA task?

2. How to design end-to-end trainable network architecture?

We hypothesize that it should, so rather than using just a single number to train the network, we take multitask learning (MTL) approach to AQA. To enable this kind of MTL approach and evaluate its performance, we first of all introduce an MTL-AQA dataset. Secondly, we propose two MTL architectures. Thirdly, through experimentation, we demonstrate that our MTL approach outperforms all the existing AQA approaches; MTL offers better generalization; there's a utility in learning AQA-specific features, rather than opting for action recognition features.

Multitask Approach to AQA

In MTL, a model is learnt such that it is suitable for serving more than one task. The tasks are generally related in nature. Since the tasks are related not completely identical, a part of the network is shared/common, which branches into heads which are specific to each individual task. The network is then trained end-to-end using the loss, which is a sum of losses pertaining to all the tasks. The common network body learns richer features, which can explain all the tasks.

Task selection : As we noted before, since the AQA is a function of *what* action was performed, and *how well* that was performed, our choice of auxiliary tasks (our main task is the prediction of action quality score) becomes very natural. In order to characterize action, we consider detailed action recognition as an auxiliary task, which would be responsible for the what action part. Generating a verbal commentary that describes good and bad points of the performance becomes an auxiliary task that handles the how well part.

Formalization : in the following, we formalize the problem settings, and objective functions.

AQA is a regression problem, therefore, AQA branch produces sum of Euclidean loss and L1 loss. We found that using L1 loss in addition to L2 loss yields better results.

For detailed action recognition, we mean detailed dive recognition in particular. A dive, as explained in detail in next section, can be broken down into five components. Detailed dive identification refers to identifying each one of these five components. Detailed adive classifica-

tion branch produces five cross-entropy losses.

Commentary generation is a captioning task. So, commentary generation branch produces negative log-likelihood loss.

The overall loss is a sum of three losses described above.

Spatiotemporal features learnt using 3D CNN's capture appearance and salient motion patterns, which make them best candidate for AQA. However, 3D CNN's is more suitable for small length clips than for long videos. So, we use 3D CNN's to compute spatiotemporal features for small clips, and then aggregate those to get whole video-level features in the following two ways. As discussed previously, MTL architectures have a common body and task-specific heads.

1. **Averaging as aggregation (C3D-AVG)**: Throughout the action, an athlete can be considered as collecting (or losing) points. This operation is an addition operation of points. Addition is a linear operation. A good metric to evaluate if learnt features are good, is that linear operations on those features become meaningful. So, we propose to use addition of clip-level features to obtain video-level features. Doing so would help the network learn good features. Since the captioning is sequence to sequence task and ordering matters, LSTM is used on top of clip-level features.

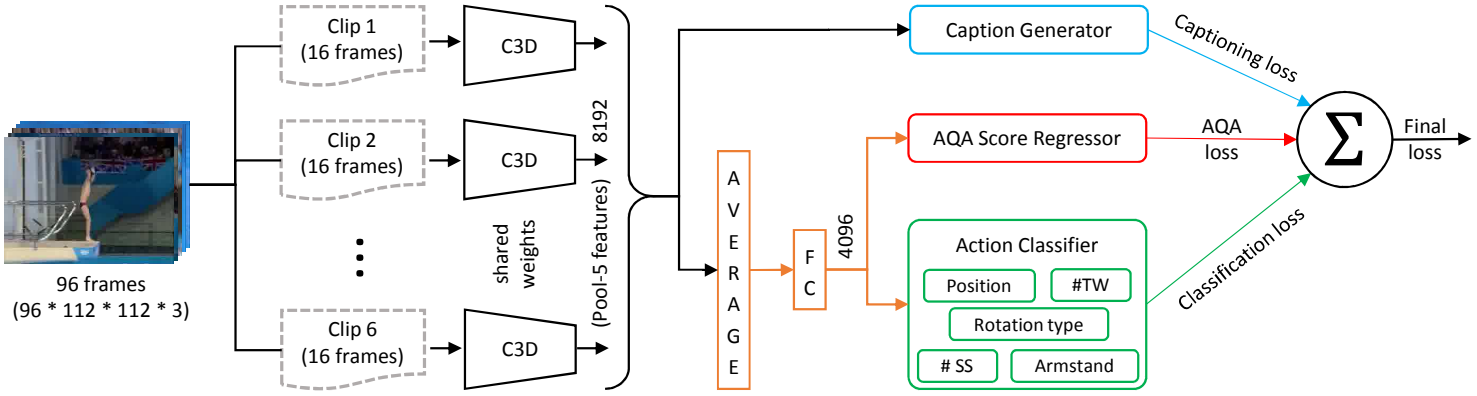


Figure 5.1: Architecture of C3D-AVG-MTL.

	(Common network body)	
	C3(32); BN MP(1,2,2) C3(64); BN MP(2,2,2) {C3(128); BN} x2 MP(2,2,2) {C3(256); BN} x2 {C3(d=2,256); BN} x2 Dropout(0.5)	
	(Task-specific heads)	
(AQA Score Head)	***(Action recognition Head)***	***(Captioning Head)***
C1(12) {Cntxt net} MP(2,2,2) C3(12); BN	C1(12) {Cntxt net} MP(2,2,2) C3(12); BN	C1(12) {Cntxt net} MP(2,2,2) C3(12); BN
C3(1) AP(2,11,11)	***(Action recognition sub-heads)***	Enc. GRU Dec. GRU

Table 5.1: MSCADC-MTL architecture. C3(d,ch): 3D convolutions, ch-no. of channels, d-dilation rate. C1: 1x1x1 convolutions. BN: batch normalization. MP(kr): max pooling operation, kr-kernel size. Cntxt net: context net for multi-scale context aggregation. AP: average pooling across (2x11x11) volume.

2. **Multiscale Context Aggregation with Dilated Convolutions (MSCADC)**: In this architecture, we downsample the action sequence from 96 frames to 16 frames by dropping frames. This way the need to aggregate clip-level features is eliminated. Inspired by the performance of the architecture used by Nibali *et al.* [23] on the task of Diving classification, we adopt their architecture for our case. The backbone is based on C3D architecture, and in addition, incorporates dilated convolutions [44] and batch normalization [13]. We also use a separate context net for each task-specific head. This architecture is fully convolutional – has lesser parameters, which allows to increase the spatial resolution of the input.

Multitask AQA Dataset

There is no dataset that has labels that are required by our MTL approach. So for approach, we first of all compile a dataset, with following three labels:

1. AQA score: which is an average of seven judges

2. Detailed dive class: a dive has five components as detailed in Table 5.3. For each dive sample, we annotate all the components.

3. Verbal commentary: before the advent of television, people used to "watch" sports events with the help of commentary. Commentary is usually delivered by retired athletes, who are experts in their own field. Commentators commentate on the good and bad points of the performance. We use Google Speech-to-Text [1] interface to transform commentary audio to text with timestamps for synchronization.

Our MTL-AQA dataset contains 1412 samples, and is much more diverse than the existing Diving datasets. In addition to filling data void in AQA, it can help researchers in other sub-fields as well.

Dataset	Events	Height	Genders	# Samples	Events	View Variation/ Background	Labels
MIT Dive [31]	Individual	10m Platform	Male	159	1	No/Same	AQA score
UNLV Dive [28]	Individual	10m Platform	Male	370	1	No/Same	AQA score
Ours MTL-AQA	Individual, Synchronous	3m Springboard, 10m Platform	Male, Female	1412	16	Yes/Different	AQA score, Action class, Commentary

Table 5.2: Details of our newly introduced dataset, and its comparison with the existing AQA datasets.

Position	Armstand	Rotation type	# SS	# TW
Free Tuck Pike	No Yes	Inward Reverse Backward Forward	0 to 4.5	0 to 3.5

Table 5.3: Classification of dives. Each combination of the presented sub-fields produces a different kind of maneuver.

Experiments

Implementation : We implement our models using PyTorch [30]. We pretrained network backbones on UCF101 dataset [34]. For captioning module (encoder and decoder), we use GRU type cell [6] with a dropout [35] rate of 0.2. Maximum caption length is set to 100 words. Vocabulary size for full dataset is 5779 words. α, β, γ are set to 1, 1, 0.01, respectively. All models are optimized using ADAM optimizer [17]. We train for 100 epochs with an initial learning rate of 0.0001. We use center crop of the video stream and apply random horizontal flipping. Further model-specific details are given in the following.

C3D-AVG : center crop of size 112x112 is cut from a video of size 171x128 pixels. All the dive samples are normalized to 96 frames.

MSCADC : center crop of size 180x180 is cut from a video of size 640x360 pixels.

Tasks	C3D-AVG	MSCADC
AQA	89.60	84.72
+ Cls	89.62	85.76
+ Caps	88.78	85.47
+ Cls + Caps	**90.44**	**86.12**

Table 5.4: STL vs. MTL across different architectures. Cls - classifiction, Caps - captioning. First row shows STL results, while the remaining rows show MTL results.

Evaluation metrics : We use Spearman's rank correlation as the performance metric for AQA task. For classification task, we use accuracy, and for commentary, we use Bleu, Meteor, Rouge, and CIDEr scores.

Single-task vs. Multitask approach In this experiment, we determine the effect of incorporating auxiliary tasks.

We observe that incorporating task helps improve performance on both the architectures. Our full model outperforms all the other models on both architectures, showing the efficacy of MTL is not just limited to an architecture. C3D-AVG outperforms MSCADC, while MSCADC has an advantage of being computationally less intensive and faster.

Additionally, we also compare with segment-aware methods. Since segment-aware methods use their annotations to identify segments, we compare our models using UNLV-Dive dataset [28]. Our C3D-AVG-MTL outperforms all the segment-aware methods as well.

Furthermore, we also evaluate and report the performance on auxiliary tasks. C3D-AVG-MTL outperforms MSCADC-MTL.

We believe that MTL yielding better than STL because it can achieve better generalization. To be more ascertain about this, we train both the models using fewer samples. We find that

Method	Sp. Corr.
Pose+DCT [31]	26.82
C3D-SVR [28]	77.16
C3D-LSTM [28]	84.89
Ours MSCADC-STL	84.72
Ours C3D-AVG-STL	**89.60**
Ours MSCADC-MTL	86.12
Ours C3D-AVG-MTL	**90.44**
Segment-specific methods (train/test on UNLV Dive [28])	
S3D (best performing in [41])	86.00
Li *et al.* [20]	80.09
Ours MSCADC-STL	79.79
Ours C3D-AVG-STL	83.83
Ours MSCADC-MTL	80.60
Ours C3D-AVG-MTL	**88.08**

Table 5.5: Performance comparison with the existing AQA approaches.

	Nibali *et al.* [23]	Ours-MTL MSCADC	Ours-MTL C3D-AVG
Position	90.78	78.47	96.32
Amstand	100.00	97.45	99.72
Rotation type	89.81	84.70	97.45
# Somersaults	86.89	76.20	96.88
# Twists	95.15	82.72	93.20

Model	B1	B2	B3	B4	M	R	C
C3D-AVG	0.26	0.10	0.04	0.02	0.11	0.14	0.06
MSCADC	0.25	0.09	0.03	0.01	0.11	0.13	0.05

Table 5.6: Performance on auxiliary tasks. Comparison with [23] on the task of dive classification is presented in the top table. Bleu(B), Meteor(M), Rouge(R), CIDEr(C) scores for the captioning task are presented in the bottom table.

# samples	1059	450	280	140
STL	89.60	77.27	69.63	64.17
MTL	**90.44**	**83.52**	**72.09**	**68.16**

Table 5.7: STL vs. MTL generalization. Training using increasingly reduced no. of training samples.

MTL consistently outperformed STL, and also the gap in the performance seems to widen with fewer training samples.

AQA-orientedness of the learned features

We trained our models end-to-end with the aim of learning better features. In the following experiment, we evaluate to see if training end-to-end was beneficial over using action recognition features.

We learn linear regressors on top of all the convolutional layers to predict the final scores. The idea behind this experiment is that if the correlation is higher as compared to that obtained with action recognition features, then it supports the practice of learning AQA-oriented features. This experiment was adopted following the experiment introduced in a work by Zhang *et al.* [45].

We carry out the experiment on our MTL-AQA Diving dataset, and observe that our C3D-

	c1	c2	c3	c4	c5
Action Recognition	71.01	71.39	73.13	76.34	73.69
Dive Recognition	72.43	70.15	70.35	57.20	37.63
C3D-AVG-MTL	**74.26**	**77.95**	**82.78**	**86.18**	**85.75**

Table 5.8: Performance of fitting linear regressors on the activations of all the convolutional layers.

AVG-MTL learns better features than Dive recognition and action recognition models at all intermediate layers.

Discussion

In this work, we introduced MTL approach to AQA, and showed that it works better than STL because of the better generalization it can provide. Ability to learn using fewer training samples is especially useful in case of AQA, where datasets are typically pretty small. We also tried to keep the hyperparameter tuning to minimum. Our best performing model, C3D-AVG-MTL achieved new state-of-the-art results with 90.44% correlation.

Although in this paper, we focused Diving dataset, we did not make any model design choices that was limited to or specific to Diving action class. Our method can be applied to other actions as long as required annotations can be obtained. In addition, our approach can also be extended to Skills Assessment domain, where an expert can break a complex action down into more primitive sub-actions, and can commentate on them as well.

CHAPTER 6

CONCLUSION

In this dissertation, we discussed three works. First work, we noted that the drawbacks of pose based features – poor pose estimation, and neglecting visual cues. To that end, we proposed to use 3D CNN representations. We surpassed state-of-the-art approaches by a wide margin. Unlike existing approaches which learn action-specific approaches, in our second work, we propose to model an all-action model, because such a model can exploit common action quality elements across various actions. All-action modeling has the benefits of being able to use data more efficiently, offer better generalization and adaptability. Noting that a single score might not be sufficient to characterize a complex action, in our third work, we propose to take multitask learning approach as opposed to existing single task learning approaches. In our MTL approach, we propose to learn AQA-oriented representations by optimizing our proposed networks end-to-end jointly for three tasks: commentary generation, detailed action identification, and AQA scoring. MTL approach surpasses all the previous approaches. In addition, our proposed all-action modeling and MTL approaches are applicable to other domains.

As AQA is an upcoming sub-field, there was data shortage. To this end, we contribute by releasing UNLV-Dive and UNLV -Gymvault datasets in our first work, followed by releasing AQA-7 dataset, and in third work, we released first-of-its-kind MTL-AQA dataset richly annotated with AQA scores, detailed commentary and action class, which may be useful for other

computer vision tasks as well.

AQA has a lot of potential, but is not gaining as much attention from the community as for *e.g.*, action recognition and detection. Some of the directions that future efforts can pursue are as follows:

- Compile further datasets
- There is a lot of potential in improving performance on AQA task
- Feedback systems that can provide suggestions to users on improving their performance would also be very helpful

BIBLIOGRAPHY

[1] Cloud speech-to-text. https://cloud.google.com/speech-to-text/. Accessed: 2019-09-30.

[2] Fdration internationale de natation. http://www.fina.org/. Accessed: 2019-09-30.

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

[4] G. Bertasius, H. S. Park, X. Y. Stella, and J. Shi. Am i a baller? basketball performance assessment from first-person videos. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2196–2204. IEEE, 2017.

[5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[6] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[7] H. Doughty, D. Damen, and W. Mayol-Cuevas. Who's better, who's best: Skill determination in video using deep ranking. *arXiv preprint arXiv:1703.09913*, 2017.

[8] H. Doughty, W. Mayol-Cuevas, and D. Damen. The pros and cons: Rank-aware temporal attention for skill determination in long videos. In *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, pages 7862–7871, 2019.

[9] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161, 1997.

[10] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

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

[12] W. Ilg, J. Mezger, and M. Giese. Estimation of skill levels in sports based on hierarchical spatio-temporal correspondences. In *Joint Pattern Recognition Symposium*, pages 523–531. Springer, 2003.

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[14] K. Jarrett, K. Kavukcuoglu, Y. LeCun, et al. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153. IEEE, 2009.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR 2011*, pages 3361–3368. IEEE, 2011.

[20] Y. Li, X. Chai, and X. Chen. End-to-end learning for action quality assessment. In *Pacific Rim Conference on Multimedia*, pages 125–134. Springer, 2018.

[21] Y. Li, X. Chai, and X. Chen. Scoringnet: Learning key fragment for action quality assessment with ranking loss in skilled sports. In *Asian Conference on Computer Vision*, pages 149–164. Springer, 2018.

[22] Z. Li, Y. Huang, M. Cai, and Y. Sato. Manipulation-skill assessment from videos with spatial attention network. *arXiv preprint arXiv:1901.02579*, 2019.

[23] A. Nibali, Z. He, S. Morgan, and D. Greenwood. Extraction and classification of diving clips from continuous video footage. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–48, 2017.

[24] R. Ogata, E. Simo-Serra, S. Iizuka, and H. Ishikawa. Temporal distance matrices for squat classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[25] A. Paiement, L. Tao, S. Hannuna, M. Camplani, D. Damen, and M. Mirmehdi. Online quality assessment of human movement from skeleton data.

[26] P. Parmar and B. Morris. Action quality assessment across multiple actions. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1468–1476. IEEE, 2019.

[27] P. Parmar and B. T. Morris. Measuring the quality of exercises. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pages 2241–2244. IEEE, 2016.

[28] P. Parmar and B. T. Morris. Learning to score olympic events. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 76–84. IEEE, 2017.

[29] P. Parmar and B. T. Morris. What and how well you performed? a multitask learning approach to action quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–313, 2019.

[30] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[31] H. Pirsiavash, C. Vondrick, and A. Torralba. Assessing the quality of actions. In *European Conference on Computer Vision*, pages 556–571. Springer, 2014.

[32] D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

[33] F. Sardari, A. Paiement, and M. Mirmehdi. View-invariant pose analysis for human movement assessment from rgb data. In *International Conference on Image Analysis and Processing*, pages 237–248. Springer, 2019.

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

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[36] M. Strickert., F. M. Schleif, U. Seiffert, and T. Villmann. Derivatives of pearson correlation for gradient-based analysis of biomedical data. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 12(37):37–44, 2008.

[37] L. Tao, A. Paiement, D. Damen, M. Mirmehdi, S. Hannuna, M. Camplani, T. Burghardt, and I. Craddock. A comparative study of pose representation and dynamics modelling for online motion quality assessment. *Computer vision and image understanding*, 148:136–152, 2016.

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Computer Vision (ICCV), 2015 IEEE Inter-*

national Conference on, pages 4489–4497. IEEE, 2015.

[39] V. Venkataraman, I. Vlachos, and P. Turaga. Dynamical regularity for action analysis.

[40] K. Wnuk and S. Soatto. Analyzing diving: a dataset for judging action quality. In *Asian Conference on Computer Vision*, pages 266–276. Springer, 2010.

[41] X. Xiang, Y. Tian, A. Reiter, G. D. Hager, and T. D. Tran. S3d: Fusing segment-level p3d for action quality assessment.

[42] C. Xu, Y. Fu, B. Zhang, Z. Chen, Y.-G. Jiang, and X. Xue. Learning to score figure skating sport videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[43] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392. IEEE, 2011.

[44] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[45] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*, 2016.

[46] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, M. A. Clements, and I. Essa. Automated assessment of surgical skills using frequency analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 430–438. Springer, 2015.

CURRICULUM VITAE

Graduate College
University of Nevada, Las Vegas

Paritosh Parmar

Email address: parmap1@unlv.nevada.edu; 08bec059@nirmauni.ac.in

Degrees:

Bachelor of Technology - Electronics and Communication Engineering, 2012
Nirma University, India

Master of Technology - Robotics, 2014
SRM University, India

Doctor of Philosophy - Electrical Engineering, 2019
University of Nevada, Las Vegas, USA

Dissertation Title: On Action Quality Assessment

Dissertation Examination Committee:

Chairperson, Dr. Brendan T. Morris, Ph.D.
Committee Member, Dr. Emma Regentova, Ph.D.
Committee Member, Dr. Venkatesan Muthukumar, Ph.D.
Graduate Faculty Representative, Dr. Mohamed Trabia, Ph.D.