CS230

Automated Figure Skating Technical Specialist: Identification of Jumps Utilizing Video Classification

Alea Delmastro
Stanford University
alead@stanford.edu

Abstract

Current judging and scoring approaches at figure skating competitions require human visualization of skills by a technical specialist in order to render a score. Factors such as visibility, time constraints, and fatigue after a long competition can impact a technical specialist's ability to accurately classify figure skating jump attempts, thus introducing unnecessary bias in scoring. US Figure Skating Association has a pressing need and interest in automating their scoring methods to eliminate such biases. Therefore, I propose a video classification algorithm built using a temporal segment network (TSN) to classify the six jump types in singles figure skating: axel, salchow, toe loop, loop, flip, and lutz. By utilizing pose estimation on a set of 2.0 second video clips illustrating jump attempts from figure skating competitors, I optimized a TSN model constructed with a ResNet-50 backbone to predict these six classes with 62.22% mean class accuracy. The model does exceptionally well at predicting jumps with forward take offs (i.e., axels) with an F1 score of 0.939; however, there is still room for improvement in classifying "edge jumps" (i.e., salchows and loops). In conclusion, this project lays the foundations for future figure skating classification algorithms to eventually distinguish inaccuracies of jump attempts (i.e., under-rotations or edge changes), thus establishing the frame-work for an automated technical specialist.

1 Introduction

In figure skating, there are six jump types - axel (3A), salchow (3S), toe loop (3T), loop (3Lo), flip (3F), and lutz (3Lz) - with different base point values as judged by a technical specialist on the judging panel. The specialist identifies the jump type, the number of revolutions completed in the air, and the accuracy of the jump. Because technical specialists are dependent on the accuracy of the human eye, clear visibility, and time constraints, questions arise about their scoring of the jumps in real time. The motivation for this project is to automate the role of the technical specialist to minimize bias and variability in scoring across competitions, nationally and internationally. The US Figure Skating Association (USFSA) has expressed interest in automating their scoring methods.

Figure skating jumps are usually classified by their temporal movement. As a result, finding one time that distinguishes each jump is challenging. In order to automate jump classification, video classification is the best approach. While convolutional neural networks (CNNs) are ideal for many different computer vision machine learning tasks, CNNs have not shown to be optimal in action recognition in videos. Conversely, a temporal segment network (TSN) utilizes a segment-based sampling and aggregation module, which enables it to learn action models effectively[1]. Therefore, to allow the model to visualize each individual jump attempt, the input to the algorithm is a 2.0 second

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

color video displaying one jump. I then use the temporal segment network to output a predicted jump type from the six jump classes: 3A, 3S, 3T, 3Lo, 3F, and 3Lz.

2 Related work

Previous implementations took a holistic approach to scoring figure skating, rather than scoring at a skill-by-skill level, as they evaluated full routines and predicted an overall score[2]. Other machine learning analyses of figure skating videos incorporated pose estimation in their assessment of the quality of skills performed rather than classifying skill types[3]. While beneficial, these approaches have limited applicability to current judging as the technical score is very precise and dependent on the specific skills executed based on a table of points determined by the International Skating Union.

One recent implementation curated a figure skating data set from competitions in the 2017 - 2018 season consisting of all the technical skills performed by singles skaters: jumps, spins, and step sequences[4]. This implementation's objective was to classify all attempted technical skills across these three categories. While their classification approach achieved remarkable performance, their data set was very small, with the largest class size containing <250 attempts, and, in their classification analyses, their data set did not contain classes for all six jumps types.

Outside of figure skating, action recognition and skill classification are thriving fields in computer vision. Firstly, deep learning approaches were applied to classify different yoga poses using pose-estimated images that were introduced into a CNN[5]. This method achieved a test set accuracy of >0.98 across its six classes, showing promise in incorporating pose estimation to jump attempts. Since figure skating jumps are dependent on their temporal movement, however, this image classification method is expected to not be as effective as video classification. Action recognition and classification also were applied in the sport of gymnastics, a sport similar to figure skating where the outcome depends primarily on performance judgments. For example, action recognition algorithms were developed for score prediction of specific skills in artistic gymnastics[6]. In this application, videos were used as input and then low-level computer vision techniques were applied on a frame-by-frame basis to construct a spatio-temporal trajectory containing details of the human motion. Similar to the holistic figure skating scoring approach mention by Xu et al, this project's goal is to predict a score and not necessarily classify skills; therefore, its objective does not align well with the role of the technical specialist. The 2021 Tokyo Olympics are expected to utilize automated gymnastics scoring methods, which show promise in applying such computer vision frameworks in similar sports[7].

3 Dataset and Features

I am using video footage (licensed from the USFSA) from 12 different competitions that took place between March 2019 and March 2020. The data set includes jump attempts from the men's and women's short and free skating programs at the junior and senior levels. In total, 94 men and 109 women from 50 countries are represented in this data set. I found the labels of jump type and accuracy by reviewing protocol sheets published online after each skating event (Supplemental Figure 1). Only triple jumps are included in the data sets for simplicity. In addition, for jump classification, only jump attempts without an "e" or "!" error will be included, since these edge errors change the jump type from a "flip" to a "lutz" or vice versa. The data set consists of 300 - 700 attempts of each jump type. The frequencies of jumps included in the data set are provided in Table 1.

Jump	**3A**	**3S**	**3T**	**3Lo**	**3F**	**3Lz**	**Sum**
Men	**389**	**153**	**291**	**184**	**196**	**309**	**1521**
Women	**41**	**178**	**385**	**279**	**275**	**284**	**1442**
Total	*430*	*331*	*676*	*463*	*471*	*593*	*2964*

Table 1: Summary of frequency of jump types across dataset by gender. Only jumps without an edge error were included. The jump classes are overall balanced; however, the one class that is not balanced by gender is the 3A category since triple axels are more infrequently executed by women due to their inherent difficulty.

For data pre-processing, before clip extraction, I vertically flipped jump attempts that were performed in the clockwise direction to eliminate class imbalance based on rotational direction and make the

Figure 1: Example frames of each jump type along with their labels in the top left corner.

Figure 2: Example output of OpenPose pose estimation on a figure skating clip.

data set consistent. During data extraction, I down-scaled the 59.94 fps clips from 1280 x 720 px to 320 x 180 px to reduce training times and make the resolution more compatible with the model architecture. Example frames from each jump type are displayed in Figure 1.

Inspired by previous figure skating computer vision studies, I incorporated pose-estimation into my data set, utilizing the publicly available OpenPose demo software on the video clips (Figure 2)[8].

Since the data set is relatively small, it was split using 70% of the videos as training data, 15% as development data, and 15% as testing data. The training set consists of 2084 jump attempts and both the validation and test sets contain 445 jump attempts. The distribution of all these sets are equal.

4 Methods

For video classification, I implemented MMAction2, an open-source toolbox for video understanding based on PyTorch[9]. It incorporates a TSN for action recognition and has a pre-trained recognizer based on the Kinetics400 dataset, a collection of large-scale, high-quality datasets of URL links of up to 650,000 video clips that cover 400 human action classes[10]. Since the pre-trained weights for this model were representative of various human actions, I considered them as an appropriate starting point for my figure skating data set. The backbone of the model is a Residual Network (ResNet) with 50 layers[11], and the head is a TSN. The network is trained using stochastic gradient descent and has a cross entropy loss function, optimal for mutli-class classification.

To prove that video classification is more appropriate for this application, I also implemented a simple CNN utilizing single frames from the clips as input data. The single frames, representative of the time when the toe pick hits the ice before take-off, were recorded while annotating the times of extraction for the video clips. The CNN had the following forward propagation architecture: CONV2D -> RELU -> MAXPOOL -> CONV2D -> RELU -> MAXPOOL -> FLATTEN -> FULLYCONNECTED. In addition, as part of the training strategy, the model included a cross entropy loss function.

5 Experiments/Results/Discussion

Before selecting video classification, I tested single frames (original and pose-estimated) extracted from the clips in the CNN described above. For the original single frame data set, the CNN had a training set accuracy of 0.932 after 100 epochs, a learning rate (LR) of 0.009, and a minibatch size of 64 and a validation set accuracy of 0.265. Next, for the pose-estimated single frame data set, with the same hyperparameters, the CNN had a training set accuracy of 1.0 and a validation set accuracy of 0.333. While the pose-estimated single frame data set improved in validation set accuracy by almost 10%, this accuracy did not meet the expectations of the project's objectives.

Next, a preliminary test of the MMAction2 pre-trained model with the original video clip data set indicated that, after only 30 epochs, the mean class accuracy for the validation set was 0.4655, almost a two-fold improvement over the accuracy of the simple CNN used for image classification. This confirmed my original assumptions about the benefits of video classification over image classification.

Once I proved video classification ideal, I began testing several hyperparameters. First, I compared validation set mean accuracy after varying epoch size, keeping all other parameters at default (LR of 0.000071825 and dropout ratio of 0.4). With 50 epochs, the original video data set improved to a mean class accuracy of 0.541; however, changing the number of epochs to 100 only improved accuracy by 0.009 with double the training time. Therefore, the optimal epoch size for future experiments is 50.

To tune the LR and dropout ratio (DR) hyperparamters, I applied a grid search on both the original and pose-estimated data sets, setting pose estimation as a third hyperparameter. For LR, I tested 0.00001, 0.000078125 (default), 0.0001, and 0.001. For DR, I tested 0.7, 0.75, 0.8, and 0.85. I selected LR and DR ranges empirically after wide searches across LR range of 0.1 and 0.00001 and DR range of 0.2 and 0.9. In total, during the grid search, I tested 32 different trained models. Provided in Supplemental Tables 1 and 2 are the mean class validation set accuracies of each model.

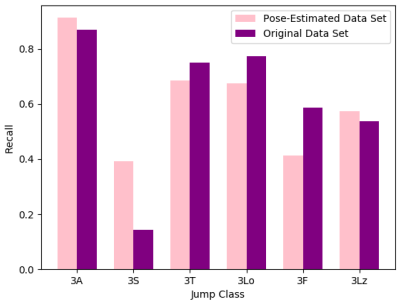


Figure 3: Comparing Recall Scores between Original and Pose-Estimated Trained Models

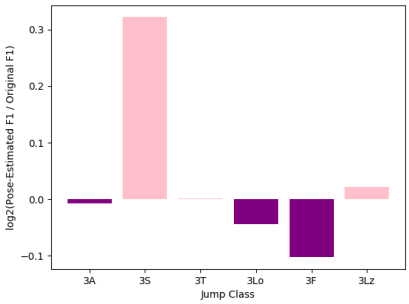


Figure 4: Log2 Ratio of F1 Scores Between Pose-Estimated and Original Trained Models

Since the objective of this project is to produce a model that can classify all the jumps accurately, I selected the best models from the grid search by comparing the validation set mean class accuracies. Both the original and pose-estimated grid searches demonstrated that models with a LR of 0.0001 and a DR of 0.8 were best, with mean accuracies of 0.6086 and 0.6101, respectively. Given that these mean accuracies are almost equivalent, to choose whether to move forward with the original or pose-estimated data set, I compared their recall and F1 scores for each class (Figures 3-4). When reviewing the produced confusion matrices for these models (Supplemental Figures 2-3), the original data set model correctly classified more 3T, 3Lo, and 3F attempts; however, the pose-estimated data set model significantly improved the recall of the 3S category, which generally throughout the models had the lowest recall as many 3S jumps were falsely characterized as 3Lo. This improvement indicates that the pose-estimated model better learned that the 3S takes off from the left foot and the 3Lo takes off from the right. By taking the log2 of the ratio of F1 scores for each class, it becomes clear that the improved F1 score of 3S with pose estimation is substantial compared to the slight decreases in F1 score of 3T, 3Lo, and 3F classes (Figure 4). As a result, the best model after hyperparameter tuning is the pose-estimated trained model with 50 epochs, 0.0001 LR, and 0.8 DR.

Applying the model on the test set produces a mean accuracy of 0.6222, a slight improvement over the validation set and significantly better than random. Given this consistency in accuracy between

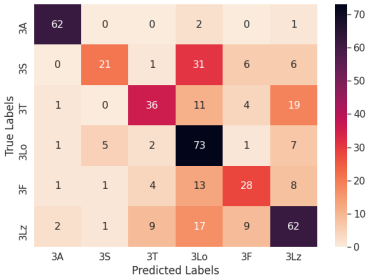


Figure 5: Confusion Matrix from Test Set

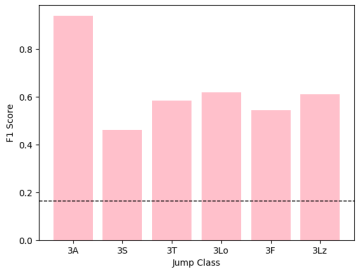


Figure 6: F1 Scores across Classes in Test Set compared to F1 score of random (dashed line)

the validation and test set, the training set does not appear to be overfitted, especially since the model showed improvement throughout the hyperparameter grid search. The confusion matrix in Figure 5 demonstrates that both the precision and recall for the 3A class are both above 0.9. This result is expected given that 3A is the only jump type with a forward take off; therefore, the model is able to distinguish forward motion from backward motion. The 3S class still maintains a relatively small recall score as the a large majority of the 3S jump attempts were classified as 3Lo; this is a reasonable error since both jumps are considered "edge jumps", where the jump does not require a tap of the toe pick to lift off the ice. The 3Lo class was the least precise overall. In addition to the falsely classifying 3S attempts as 3Lo, it also misclassified 3T, 3F, and 3Lz attempts as 3Lo. Given that both 3F and 3Lz attempts also take off from the right foot, this misclassification is somewhat reasonable; but it is surprising that some 3T attempts were predicted as 3Lo since 3T takes off from the other foot and is considered as a "toe jump". Surprisingly, the model did well in not misclassifying 3F as 3Lz and vice versa. Both of these jumps are "toe jumps" and require the skater to tap with the right foot - the only difference is the edge of the take off; this implies that the model was able to adequately learn different edge angles at the take off. The resulting F1 scores for each class are displayed in Figure 6 compared to random classification. We can see that overall each individual class had a significantly greater F1 score than the F1 score expected at random.

6 Conclusion/Future Work

In conclusion, I optimized a TSN model constructed with a ResNet-50 backbone to predict six classes - 3A, 3S, 3T, 3Lo, 3F, and 3Lz - with 62.22% mean class accuracy. The model does exceptionally well at predicting jumps with forward take offs (i.e., axels) with an F1 score of 0.939; however, there is still room for improvement for classifying "edge jumps" (i.e., salchows and loops). With pose estimation, the model learned how to distinguish the left and right leg with more accuracy, which is important in discerning jump type. In addition, the model reasonably learned the different edge angles to distinguish lutzes from flips and vice versa.

Given more time and access to additional competition footage, firstly, future models would benefit from more training data. While 2964 video clips can be considered substantial, it was not sufficient to achieve a significantly high accuracy across all classes. Secondly, the data set could also become more balanced. Despite a relatively uniform distribution across the classes, the smallest class (3S) still had only half the number of jump attempts as the largest class (3T). Lastly, increasing the depth of the ResNet backbone from 50 to 101 layers should significantly improve the mean accuracy. Presently, there are no available pre-trained weights for an optimized video action recognition ResNet-101.

Once jump classification becomes more accurate, I would classify jump inaccuracies. When scoring, technical specialists also need to determine jump errors: whether a jump is fully rotated and whether the jump has a clean edge take-off for flips and lutzes only. This form of classification would utilize a similar data set with more attention to detail. These kinds of errors are more infrequent; therefore, there would be significant class imbalance necessitating additional data curation.

This project demonstrates the potential of computer vision solutions in the realm of figure skating to modernize the sport and eliminate biases during competition scoring by technical specialists.

7 Contributions

I completed this whole project by myself, including the manual annotation of the videos for clip extraction, hyperparameter tuning, and final result analyses. I would like to thank USFSA for its provision of the video footage.

References

[1] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L. V. (2019). Temporal Segment Networks for Action Recognition in Videos. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(11), 2740-2755. doi:10.1109/tpami.2018.2868668.

[2] C. Xu, Y. Fu, B. Zhang, Z. Chen, Y. -G. Jiang and X. Xue, "Learning to Score Figure Skating Sport Videos," in IEEE Transactions on Circuits and Systems for Video Technology, 30(12), 4578-4590, Dec. 2020, doi: 10.1109/TCSVT.2019.2927118.

[3] Pirsiavash H., Vondrick C., Torralba A. (2014) Assessing the Quality of Actions. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, 8694. Springer, Cham. https://doi.org/10.1007/978-3-319-10599-4$_3$6.

[4] Liu, S., Liu, X., Huang, G., Feng, L., Hu, L., Jiang, D., ... Qiao, H. (2020). FSD-10: a dataset for competitive sports content analysis. arXiv preprint arXiv:2002.03312.

[5] Kothari, S. (2020). Yoga Pose Classification Using Deep Learning.

[6] Díaz-Pereira, M. P., Gomez-Conde, I., Escalona, M., Olivieri, D. N. (2014). Automatic recognition and scoring of olympic rhythmic gymnastic movements. Human movement science, 34, 63-80.

[7] Morgan, L. (2016, May 25). Gymnastics at Tokyo 2020 could feature fully-automated scoring. Retrieved from https://www.insidethegames.biz/articles/1037784/gymnastics-at-tokyo-2020-could-feature-fully-automated-robotic-scoring

[8] Cao, Z., Hidalgo, G., Simon, T., Wei, S., Sheikh, Y. (2021). OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1), 172-186. doi:10.1109/tpami.2019.2929257.

[9] Open-Mmlab Contributors. (April, 2021). Open-mmlab/mmaction2. Retrieved from https://github.com/open-mmlab/mmaction2.

[10] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., ... Zisserman, A. (2017). The kinetics human action video dataset. arXiv preprint arXiv:1705.06950.

[11] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[12] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

[13] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32 (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[14] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[15] Chollet, F. (2015). keras.

[16] Zulko (2017). MoviePy [Computer software].

Supplemental Figures

Pose-Estimated Data Set		Dropout Ratio			
	Mean Class Accuracy	**0.7**	**0.75**	**0.8**	**0.85**
Learning Rate	**0.00001**	0.2709	0.262	0.2549	0.2491
	0.00078125	0.5425	0.5262	0.5394	0.5559
	0.0001	0.6066	0.6052	0.6086	0.5579
	0.001	0.4697	0.3968	0.3175	0.1892

Supplemental Table 1: Resulting mean class accuracies of the validation set across models with the above values for learning rate and dropout ratio trained with the pose-estimated data set.

Original Data Set		Dropout Ratio			
	Mean Class Accuracy	**0.7**	**0.75**	**0.8**	**0.85**
Learning Rate	**0.00001**	0.2936	0.3046	0.3001	0.2955
	0.00078125	0.5620	0.5744	0.5616	0.5393
	0.0001	0.6015	0.6031	0.6101	0.5456
	0.001	0.4247	0.3821	0.2401	0.1868

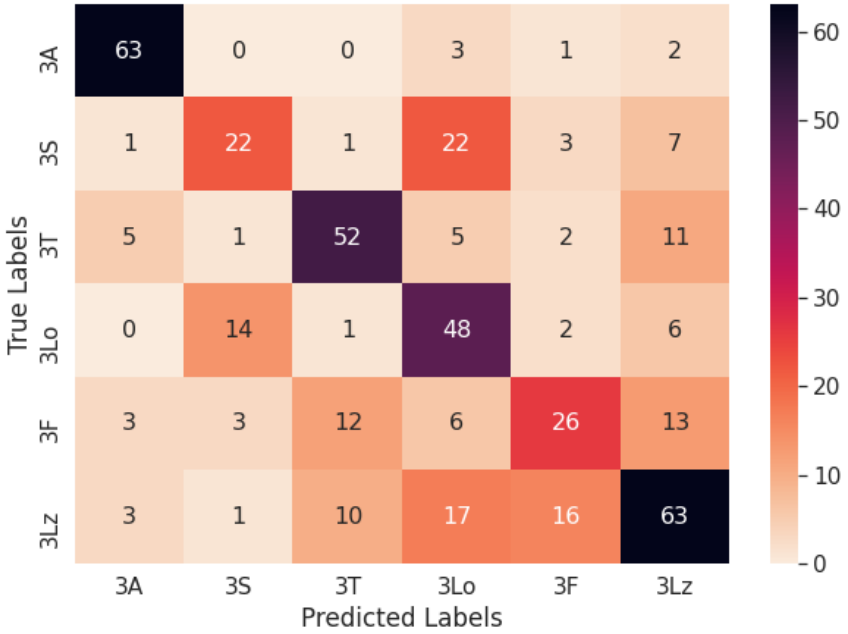
Supplemental Table 2: Resulting mean class accuracies of the validation set across models with the above values for learning rate and dropout ratio trained with the original data set.

Rank	Name	Nation	Starting Number	Total Segment Score	Total Element Score	Total Program Component Score (factored)	Total Deductions
1	Alysa LIU	USA	30	138.80	80.14	59.66	-1.00

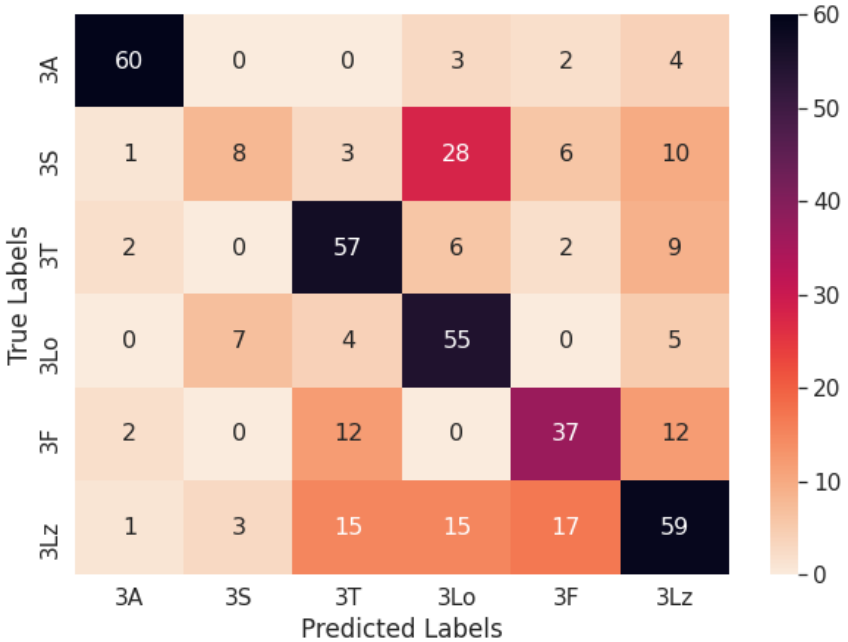
#	Executed Elements	Info	Base Value		GOE	J1	J2	J3	J4	J5	J6	J7	J8	J9	Ref.	Scores of Panel
1	3A+2T		9.30		1.26	3	2	2	1	0	0	2	2	2		10.56
2	4Lz		11.50		2.30	3	2	2	1	1	3	2	2	2		13.80
3	3A<<	<<	3.30		-1.65	-5	-5	-5	-5	-5	-5	-5	-5	-5		1.65
4	3Lo		4.90		0.84	3	1	3	1	2	2	1	1	2		5.74
5	FCSp4		3.20		1.19	2	4	4	4	4	4	3	3	4		4.39
6	StSq3		3.30		1.04	4	3	4	3	3	3	3	3	3		4.34
7	3Lz+3T		11.11	x	1.35	3	2	2	1	2	3	2	2	3		12.46
8	3Lz+1Eu+3S		11.77	x	0.84	2	1	1	1	1	2	2	1	3		12.61
9	3F!	!	5.83	x	0.00	0	0	0	0	0	-1	0	0	0		5.83
10	CCoSp4		3.50		1.25	4	4	4	2	4	3	3	3	4		4.75
11	LSp4		2.70		1.31	5	5	5	5	5	4	4	5	5		4.01
			70.41													**80.14**
	Program Components				**Factor**											
	Skating Skills				1.60	8.00	6.75	8.00	7.75	7.00	8.25	7.50	7.25	7.00		7.50
	Transitions				1.60	7.50	6.25	7.75	7.25	6.75	7.75	7.25	7.00	6.75		7.18
	Performance				1.60	7.75	7.00	8.25	7.50	7.25	7.75	8.00	7.50	7.25		7.57
	Composition				1.60	7.75	7.25	8.25	7.50	7.25	7.75	7.50	7.50	7.25		7.50
	Interpretation of the Music				1.60	7.50	7.25	8.00	7.25	7.50	7.75	7.75	7.50	7.50		7.54
	Judges Total Program Component Score (factored)															**59.66**
	Deductions:		Falls		-1.00	(1)										**-1.00**

<< Downgraded jump x Credit for highlight distribution, base value multiplied by 1.1 ! Not clear edge

Supplementary Figure 1: An example protocol sheet. The characters under the "Executed Elements" column denote the element performed and if there was an error.



Supplementary Figure 2: Validation set Confusion Matrix of Pose-Estimated Model



Supplementary Figure 3: Validation set Confusion Matrix of Original Model