Motion Classification Algorithm Comparison

David Ralph, University of Portsmouth

1 Introduction

Motion and gesture recognition is an active area of research in computer vision, and holds promise for applications in a number of areas, including human-computer-interaction, medicine, security, and robotics, as well as many others. Extensive research has taken place into means of extracting information from video for the purpose of analysis and training of models to enable accurate prediction and classification of the actions taking place.

The SIFT (Scale-invariant feature transform) algorithm, published in 2004[1] marked the start of a large growth in interest in this area, resulting in a large increase in the number of solutions proposed for this research area. Since then many other feature descriptors and descriptor extractors have been published, promising improved sets of output features (allowing for more accurate classifiers) and significant speed improvements. Such algorithms include SURF (Speeded up robust features) and FAST (Features from Accelerated Segment Test), both published in 2006, and BRISK (Binary Robust Invariant Scalable Keypoints) and ORB (Oriented FAST and Rotated BRIEF) published in 2011.

Computer vision software libraries and frameworks, such as OpenCV, commonly include a number of these algorithms for use in research and software development. This project will compare the performance and accuracy of several combinations of the available algorithms and provide analysis on the possible interactions from combining descriptor extractors with varying feature detectors. To evaluate these performance metrics, machine learning models will be trained using the output descriptors and tested by classifying human motion in videos from multiple datasets.

2 Approach

The solution consists of five key areas: feature extraction, motion representation, feature representation, training, and verification. Several technologies and algorithms exist which would be suitable for each stage, some of which are compared in the results section.

OpenCV, the library used to create the program, exposes common interfaces for descriptor matchers, descriptor extractors, and feature detectors. These interfaces make it simple to programmatically select one of a collection of several well-established and state-of-the-art algorithms. A complete list of supported algorithms can be found in the OpenCV documentation [2][3].

This program uses the common interfaces to compare the accuracy and performance of various combinations of the supported algorithms under the same operating parameters. The results of these tests make it possible to analyse the behaviour of these algorithms, and of the other training methods employed, in situations not thoroughly covered in existing literature.

The program first reads the input files, and for each generates an MHI (Motion History Image), and stores the ground truth classification of the video. MHIs are an effective method of capturing the movement from a video in a single image. This is done using image subtraction to obtain a silhouette of any regions that move between frames. This representation of the difference between frames is an example of a Motion Energy Image (MEI), one of which is shown in Fig 1.

Fig. 1. A Motion Energy Image of a person jogging

To allow for meaningful representation of a sequence of movement, an MHI can be built up incrementally from many MEIs, with the progression of time being preserved by reducing the intensity of earlier MEIs such that the greater the intensity, the more recent (later in the video) the movement. Fig 2 is an example of an MHI.

Fig. 2. A Motion History Image of a person jogging.

Once an MHI has been generated, the chosen feature detector is used to identify keypoints in the image, and then the descriptor extractor converts them into a feature representation that can be added to a Bag of Visual Words (BoW). The BoW is used to generate a histogram for each image which can be used as a descriptor to train a statistical or machine learning model. An MHI with keypoints can be seen in Fig 3.

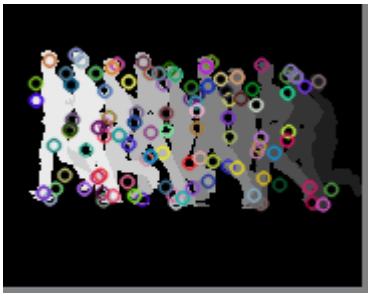

Fig. 3. A Motion History Image of a person jogging annotated with keypoints identified using SIFT.

In this project, a Support Vector Machine (SVM) is used to classify samples. This is trained using the descriptor generated by the BoW. SVMs are highly configurable, and the kernel used can have a significant impact on the complexity of the decision surface [4]. Other methods, such as neural networks could also be used to train a model from the descriptor, which may offer advantages when handling larger datasets (at the cost of added complexity) [5].

To provide useful metrics for comparison, two approaches are taken independently in this project to generate a model and make predictions. The first, referred to here as the 'Ideal model', is an SVM created using all the input videos. This is then tested against all the videos to determine the 'Ideal Accuracy'. While this is not a meaningful test of the model's generality (as it has already seen the sample it is predicting), misclassification may still occur, potentially indicating limitations of the model.

The second approach is using k-fold cross-validation to meaningfully train and test the model. This works by randomly assigning data items to folders until each is equal in size, and then training several models using all but one of the folders as training data, and the remaining one for testing. This allows all items in the dataset to be tested without including them in the training set.

After this process is complete, results are output to the console and logged in files for analysis. Various data representations from throughout the process are also exported to files in the output folder.

The program is set-up in such a way that multiple configurations can be run in sequence (within a single execution of the program), and that the program will not halt if a test fails (the test will be aborted, but other tests will continue to run). These features make it easy to run large numbers of tests automatically, and mitigate some of the issues involved in running multiple long tests.

3 Results

Table 1 (in the appendix) shows the results of several tests run using various feature detectors over the SIFT and SURF descriptor extractors. All tests were conducted using the FLANN descriptor matcher. The tests were also performed using a Brute-Force descriptor matcher, producing similar results (although all with somewhat greater run-time); the results have been omitted here as the trends remain the same. The full experiment log can be found in APPENDIX REF.

All tests used the WEIZMANN dataset [6], a set of 70 videos, each 1 to 5 seconds in length, showing one of 10 simple motions, with 7 videos showing each motion. The cross-validation split the data into 10 folders, of 7 items each.

The 'Ideal Accuracy' is the percentage accuracy when an SVM is both trained and then tested with the entire dataset, showing the upper bound on possible accuracy for cross-validation, this is explained in more detail in the Approach section.

'Total Time' includes the time to create the MHIs and BoW, as well as train and then test the 'Ideal' SVM, and perform all steps of the cross-validation.

Some of the confusion matrices produced during these tests are given analysed in the Analysis of Misclassifications section below.

3.1 Comparison of Features

It can be seen from the test results that the SURF extractor is significantly (approximately a factor of four) faster than the SIFT extractor in corresponding test cases. Both extractors average around the same level of accuracy overall, however, the distribution of accuracy values for SURF is greater, with a range of 18.5% (ideal), 31.4% (cross-validation) verses 15.7% (ideal), 21.4% (cross-validation) for SIFT. This may be due to optimisations in the SURF extractor for keypoints of a particular nature, as is evidenced by the SURF extractor giving the best performance in combination with the SURF detector, whereas the SIFT extractor shows less preference.

It is also notable that the time taken for the extractor-detector combinations of SIFT-SURF and SIFT-ORB is much greater than in other tests, this may be due to the nature of the keypoints extracted, as SURF can produce keypoint of variable (potentially large) size, and ORB, being a binary detector, will typically produce a larger number of keypoints. This then requires additional processing when extracting descriptors. An apparent possibility is that the SURF extractor is well optimised for this, whereas SIFT can be seen to give poor performance under these conditions.

The significantly greater speed of the SURF extractor, especially when using the SURF detector, would make it more suitable for use in real-time systems. Another feature of SURF, not used in these tests, is the ability to disable the orientation invariance feature of the algorithm, which results in significant speed improvements [7] in situations when orientation invariance is not required (for example static cameras).

3.2 Comparison of datasets

The program was additionally tested with another dataset. The KTH dataset [8] is a collection of 600 videos, 8 to 60 seconds in length. The videos feature various backgrounds and often contain some camera movement. The dataset features 6 classifiable actions, performed four times each by 25 people. Table 2 (in the appendix) shows the results of tests on this dataset; the configuration is otherwise the same as in the previous tests.

Both algorithms tested produced a similar level of accuracy, and SURF remained significantly faster than SIFT (as in the previous tests). Due to the additional complexities in this dataset (higher variation in video length, unstable camera, etc.), a lower accuracy is to be expected. It is notable that the Ideal Accuracy is much lower than in tests with a smaller, less complex dataset; this may be due to limitations in the representation of the data, meaning that many items cannot be correctly classified with this model due to inability to create a sufficient classifier with this volume of data.

The cross-validation accuracy is comparable to that of the tests on the WEIZMANN dataset despite the additional complexities. This is likely the result of the orientation invariant property of the algorithms used, and the ability of the model to generalise. The much greater runtime of these tests demonstrates the desirability of time efficient algorithms, as even in non-real-time systems, the time to process a large dataset is considerable.

3.3 Analysis of Missclassificaitons

Table 3 shows the confusion matrix output from the test on the WEIZMANN dataset using SURF both as the extractor and detector. This has been chosen as an example as it features the greatest accuracy, making outliers more noticeable. A text file containing a full log of all the confusion matrices produced should be supplied with this docuemnt.

		Predicted									
Actual		**2**	0	0	2	0	0	0	0	0	0
		0	**3**	0	0	0	0	0	0	0	6
		1	2	**4**	2	0	0	0	0	0	0
		3	0	1	**0**	0	0	0	0	0	3
		0	0	0	0	**7**	0	0	0	0	0
		0	0	0	0	0	**3**	2	0	0	0
		0	0	0	0	0	2	**4**	0	0	0
		0	0	0	0	0	0	0	**8**	0	0
		0	0	0	0	3	0	0	0	**7**	0
		0	2	0	1	0	0	0	0	0	**2**

Table 4. Cross-validation results from the WEIZMANN dataset using SURF.

Some classification classes were recognised much more consistently than others, with the fourth class no having any successful classifications. The motion described by this class, is a person skipping sideways while facing the camera. It is most consistently misclassified as classes 1 and 10, sideways walking and sideways hopping respectively. These actions a relatively similar in terms of movement within the frame, which is likely the cause of the misclassification. Other items in the dataset, such as class 8, which is star-jumps and the most accurately predicted, are much more distinct in terms of range of motion, making them easier to classify.

Table 4 shows the same test (SURF-SURF) on the KTH dataset.

		Predicted					
Actual		**23**	40	18	4	9	4
		24	**55**	5	2	5	4
		28	26	**27**	7	8	1
		2	1	8	**29**	24	42
		16	2	11	10	**32**	22
		5	2	3	19	5	**67**

Table 4. Cross-validation results from the KTH dataset using SURF.

In this test, the most misclassified action is class 1, boxing, whereas the most successful by a large margin is class 6, walking. It is notable that in this dataset, walking is not in any particular direction, with actors moving from any edge of the camera to the other. This gives a much greater variety descriptors for comparison, which may help make it more distinct during training compared to actions where the actor is in a fixed position such as boxing, which is most frequently misclassified as clapping (class 2), where this is also the case.

4 Conclusion

The program provides a convenient means of testing a large number of descriptor-detector combinations, works with multiple datasets, and is robust in execution (able to continue after failed tests). While the classification method does not produce good accuracy in most cases, an struggles particularly with large datasets, it is sufficient to allow comparison of the feature identification processes supported by the OpenCV common interface.

A further extension of this project would be to implement a classifier using deep neural networks, or another appropriate means of learning, to construct better models for comparison, with greater accuracy. The limitations of the SVM classifiers produced by the current program may limit the accuracy of predictions even with a good set of feature descriptors.

Further tests could include comparison of non-rotation invariant algorithms to measurably demonstrate the trade-off between runtime and accuracy over different datasets. Modification of other parameters to the detection, description, or training algorithms could also

yield interesting results showing the relationships of these components and the effects on the data output from each stage.

REFERENCES

[1] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[2] "Common Interfaces of Descriptor Extractors — OpenCV 2.4.13.2 documentation", Docs.opencv.org, 2017. [Online]. Available: http://docs.opencv.org/2.4/modules/features2d/doc/common_interfaces_of_descriptor_extractors.html#descriptorextractor-create. [Accessed: 22- Apr- 2017].

[3] "Common Interfaces of Feature Detectors — OpenCV 2.4.13.2 documentation", Docs.opencv.org, 2017. [Online]. Available: http://docs.opencv.org/2.4/modules/features2d/doc/common_interfaces_of_feature_detectors.html#featuredetector-create. [Accessed: 22- Apr- 2017].

[4] "RBF SVM parameters — scikit-learn 0.18.1 documentation", Scikit-learn.org, 2017. [Online]. Available: http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html. [Accessed: 22- Apr- 2017].

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.

[6] "Actions as Space-Time Shapes", Wisdom.weizmann.ac.il, 2017. [Online]. Available: http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html. [Accessed: 28- Apr- 2017].

[7] "Feature Detection and Description — OpenCV 2.4.13.2 documentation", Docs.opencv.org, 2017. [Online]. Available: http://docs.opencv.org/2.4/modules/nonfree/doc/feature_detection.html#int upright. [Accessed: 22- Apr- 2017].

[8] "Recognition of human actions", Nada.kth.se, 2017. [Online]. Available: http://www.nada.kth.se/cvap/actions/. [Accessed: 20- Apr- 2017].

Descriptor Extractor	Feature Detector	Ideal Accuracy (%)	Cross-Validation Accuracy (%)	Total Misclassifications (out of 70)	MHI Generation Time (ms)	Total Time (ms)
SIFT	SIFT	81.4	41.4	41	8,314	17,220
SIFT	SURF	72.9	32.9	47	28,683	67,982
SIFT	ORB	81.4	44.3	39	43,465	90,771
SIFT	BRISK	84.3	47.1	37	4,764	13,658
SIFT	FAST	88.6	54.3	32	4,696	13,181
SURF	SIFT	81.4	40.0	42	4,053	9,188
SURF	SURF	91.4	57.1	30	2,622	8,776
SURF	ORB	80.0	54.3	32	4,743	14,879
SURF	BRISK	72.9	28.6	50	1,315	6,321
SURF	FAST	77.1	25.7	52	1,084	5,786

Table 1. A table of results from the WEIZMANN dataset.

Descriptor Extractor	Feature Detector	Ideal Accuracy (%)	Cross-Validation Accuracy (%)	Total Misclassifications (out of 699*)	MHI Generation Time (ms)	Total Time (ms)
SIFT	SIFT	47.2	41.6	350	195,621	304,508
SIFT	SURF	48.1	48.7	307	380,221	582,172
SIFT	BRISK	53.4	51.1	293	139,180	238,559
SIFT	FAST	50.3	49.9	300	239,001	375,891
SURF	SIFT	36.2	33.6	398	77,500	170,501
SURF	SURF	42.6	40.4	357	68,094	164,678
SURF	BRISK	42.9	38.2	370	59,953	151,216
SURF	FAST	34.7	34.6	392	62,688	168,774

Table 2. A table of results from the KTH dataset. One video file in the collection contained encoding errors and had to be excluded.