Robust Human Action Recognition via Long Short-Term Memory

Alexander Grushin, Derek D. Monner, James A. Reggia, and Ajay Mishra

***Abstract*—The long short-term memory (LSTM) neural network utilizes specialized modulation mechanisms to store information for extended periods of time. It is thus potentially well-suited for complex visual processing, where the current video frame must be considered in the context of past frames. Recent studies have indeed shown that LSTM can effectively recognize and classify human actions (e.g., running, hand waving) in video data; however, these results were achieved under somewhat restricted settings. In this effort, we seek to demonstrate that LSTM's performance remains robust even as experimental conditions deteriorate. Specifically, we show that classification accuracy exhibits graceful degradation when the LSTM network is faced with (a) lower quantities of available training data, (b) tighter deadlines for decision making (i.e., shorter available input data sequences) and (c) poorer video quality (resulting from noise, dropped frames or reduced resolution). We also clearly demonstrate the benefits of memory for video processing, particularly, under high noise or frame drop rates. Our study is thus an initial step towards demonstrating LSTM's potential for robust action recognition in real-world scenarios.**

I. Introduction

In recent years, the long short-term memory (LSTM) neural network [11] has received increasing recognition as a general-purpose sequence processing mechanism. A key component of the LSTM architecture is a layer of *memory blocks*, where specialized *gate* neurons precisely determine what portions of a sequence are remembered, when the memories affect the network's outputs, and how long they persist. In contrast with simple recurrent neural networks (e.g., [12]), LSTM is thus able to maintain observations in memory for extended periods of time, and to effectively make use of temporally-distributed information during both training (via a specialized variant of error backpropagation) and deployment. Consequently, LSTM has been finding application towards processing long sequences of data that arise in a diverse range of tasks, such as the prediction of human movement patterns [15], the classification of speech [5] and the recognition of human actions in video [1][2][3][4], which is the focus of the present effort.

Manuscript received February 27, 2013. This work was supported by the Office of the Secretary of Defense / Office of Naval Research contract #N00014-12-M-0397. The views, opinions, and/or findings contained in this article are those of the authors, and should not be interpreted as representing the official views or policies, either expressed or implied, of the aforementioned agencies or the Department of Defense.

A. Grushin and A. Mishra are with Intelligent Automation, Inc., 15400 Calhoun Drive, Suite 400, Rockville, MD 20855, USA (corresponding author: A. Grushin; phone: 301-294-5224; fax: 301-294-5201; e-mail: agrushin@i-a-i.com).

D. D. Monner and J. A. Reggia are with the Department of Computer Science, University of Maryland, College Park, MD 20742, USA.

Given the proliferation of available video data, the *action recognition problem* now carries significant practical relevance in numerous domains. For example, the problem arises in the context of the US Navy's goal of achieving persistent intelligence, surveillance, and reconnaissance (ISR) capabilities for expeditionary warfare. Here, there is a need for the onboard/embedded processing of electro-optical (EO) and infra-red (IR) data in real-time, under low resolutions (due to poor optics), in order to extract and transmit only salient/relevant actions to Marines for tactical surveillance. Depending on the context, such actions can be *human* (e.g., a person running) or *non-human* (e.g., a vehicle making a U-turn); coordinated actions in space and time are called *activities*. A 2008 literature review [20] provides a mapping of the action recognition and activity recognition problem space, and surveys a diverse taxonomy of methods for addressing its various aspects; however, in spite of extensive research, these problems remain difficult, in the general case. Since the review was published, some promising results have emerged from the application of LSTM to the action recognition problem [1][2][3][4], yielding accurate discrimination between different action classes. However, the reported experiments were conducted under somewhat restricted conditions, where large training sets were used, where the network had the opportunity to observe an entire video sequence prior to making a decision, and where the videos were of relatively high quality. Such conditions are not guaranteed to exist in real-world settings, where *robustness* is a key requirement.

In this paper, we present initial steps towards addressing this gap. As a starting point, we have followed an experimental methodology similar to past work [2], where LSTM is applied to the well-known KTH action recognition dataset, capturing six actions performed by human subjects (URL: http://www.nada.kth.se/cvap/actions/). Each video is first preprocessed via a feature extraction technique, which extracts *histograms of optical flow* around local regions of high spatiotemporal variation. A time-ordered sequence of histograms (from some video) is then provided to the LSTM network, which classifies the action depicted in the video. Through minor variations on the LSTM model reported in [2], we were able to achieve a classification accuracy of 90.7%; this figure is slightly below the best known result for the given dataset and feature set [21], but was achieved without further preprocessing the features (unlike in [21]). Given this baseline result, we then systematically reevaluated our approach under perturbed conditions. First, we exponentially reduced the size of the training set (while maintaining the same test set), and showed that performance degrades gracefully, so long as the training videos depict actions performed by more than one human subject.

Subsequently, we reduced the duration of the input sequence that was available to the LSTM network, and demonstrated that good performance can be achieved after observing data from less than one second of video, in the average case. Finally, we have examined the effects of three forms of video quality degradation: the removal of video frames, the injection of "salt and pepper" noise, and the reduction of frame resolution. We have found that even when frame drop rates, noise rates and frame scaling factors are severe, classification performance remains relatively robust. At the same time, we established that memory is indeed beneficial to LSTM's performance on the classification task, allowing the present input to be considered in the context of past inputs. This benefit is particularly evident when high noise levels or frame drop rates are present in the video. We thus postulate that trainable memory control is essential to robust performance, since the network can learn what input information is most important to remember for effective classification, under a variety of conditions.

The focus of our paper thus encompasses the following key aspects: the LSTM technique, the robustness property and the action recognition problem. To our knowledge, all three aspects have not been addressed simultaneously. In Section II, we briefly outline past work that covered pairs of aspects: the robustness of LSTM (for other problems), the application of LSTM to action recognition, and robust action recognition (via other techniques). In Section III, we discuss our experimental methodology, with descriptions of the dataset, the feature extraction approach, and the LSTM model. In Section IV, we begin by presenting and examining a baseline performance result (achieved under somewhat idealized experimental conditions), and subsequently show how performance is affected by reductions in training set size, input sequence size and video quality. Finally, in Section V, we provide a discussion of the results, and suggest directions for future research.

II. Related Work

A. The Robustness of LSTM

There is evidence to suggest that long short-term memory is robust to various forms of degradation. Notably, the original LSTM paper [11] demonstrated that it can handle noise in both input data sequences and target output values (which are used during training). In [5], it was shown that LSTM outperformed a hidden Markov model (HMM) on a speech recognition task, even though the former utilized only one half of the training set available to the latter. It can also be postulated that because a LSTM network accepts input data as a sequence (rather than in a single batch), this data need not be complete, and the network may thus be able to make decisions under tight temporal constraints (where a decision must be made before the entire sequence becomes available). Still, at present, there are few studies that have rigorously evaluated the robustness of LSTM under the aforementioned (and other) types of degradation.

B. The Application of LSTM to Action Recognition

Over the past several years, researchers at CNRS/INSA de Lyon have produced several publications on action recognition via LSTM. In an initial study, a LSTM network was utilized to classify actions (such as goal kick, throw in, etc.) in soccer videos [1]. As input, the network was provided with pre-extracted features, namely, "bag of words" descriptors for each frame and/or the dominant motion in the scene. In subsequent research, the approach was extended by replacing the prespecified feature extractors with a deep (many-layered) convolutional neural network, which automatically learned (in a supervised fashion) to extract a set of features most appropriate to the problem [2]. This convolutional neural network was capable of performing classification on its own, but the additional use of a LSTM network yielded greater accuracy (since LSTM is better capable of dealing with temporal information). The resulting purely neurocomputational approach was applied to the KTH action recognition dataset, and achieved strong classification results; in particular, the features extracted by a convolutional neural network were found to be more effective than histograms of optical flow, which were provided to the LSTM in a baseline experiment. (Due to their simplicity, we nonetheless use such histograms as features in the present effort, because our main goal is not to achieve maximum performance under more favorable experimental conditions, but to measure the relative loss in performance under degraded conditions). In [3], the convolutional neural network was replaced with an auto-encoder, which learned to extract features in an unsupervised fashion (i.e., without labels). This approach yielded somewhat lower, but nonetheless impressive classification accuracies, which have been improved in [4]. The studies did not cover the effects of training set size, input sequence size (i.e., time available for decision making) and video quality upon classification performance.

C. Robust Action Recognition

Robustness is a key goal in action recognition research, albeit one that has not been fully reached. Here, we discuss several studies that captured measures of robustness most similar to our own. A recent thesis [22] has explored the effects of training set size upon the performance of a novel technique called sparse representation-based classification (SRC), as well as the well-known support vector machine (SVM) method, on the KTH action recognition dataset. Both approaches exhibited relatively graceful degradation under decreasing training set size, with SRC being somewhat more robust. The feature extraction approach was somewhat different from the one followed herein, yielding both histograms of optical flow and histograms of oriented gradients. In [17], it was shown that if the feature set includes local shape descriptors (which we do not consider), then a SVM is able to accurately recognize an action from just a few frames of video (once again, the KTH dataset was used), thus exhibiting robustness to the available sequence size. Another study examined the performance of the SVM under various forms of video quality degradation, namely, Gaussian blur, noise addition, lighting changes, filtering, compression, simultaneous changes in scale/rotation and frame loss [18]. Here, the FeEval dataset was utilized [19],

and various feature extractors were considered. For the feature extraction approach most similar to our own (albeit one that yielded both histograms of oriented gradients and histograms of optical flow), it was found that performance can deteriorate significantly as noise rates and frame drop rates increase. We note that the noise and frame loss models appear to have been somewhat different from those utilized in our work; also, the effects of resolution were not examined independently; instead, rescaling was applied in combination with rotation, and this was found to be highly detrimental to performance. Generally, due to significant differences in experimental procedures between our experiments and those cited herein (e.g., different datasets and/or input features were utilized), a precise quantitative comparison of robustness results would not be meaningful. Instead, where appropriate, we qualitatively contrast our results (in Section IV) with those of past work.

III. Methods

A. Dataset

Fig. 1. Snapshots of two videos from the KTH dataset: a subject boxing outdoors (left); a different subject running indoors (right).

For evaluation purposes, we utilized the aforementioned KTH dataset (Fig. 1), where each video depicts a single individual (subject) who performs an action in some scenario, at a resolution of 160 × 120 pixels. The dataset encompasses 25 distinct subjects, 6 action classes (boxing, hand clapping, hand waving, jogging, running and walking) and four scenarios (outdoor, outdoor with different zoom levels, outdoor with a different set of clothes worn by a given subject, and indoor). Because one subject-action-scenario combination is missing from the dataset, there are $25 \times 6 \times 4 - 1 = 599$ video sequences. Additionally, because every sequence captures an action that is performed multiple times, it is further split into three or (typically) four subsequences, yielding a total of 2391 shorter videos; the resulting dataset is sometimes referred to as KTH2 (e.g., in [2]).

In our experiments, we randomly partitioned the dataset into a training set and a test set by subject, with the test set containing 9 subjects, and the training set consisting of 16 or fewer subjects (depending on the experiment). Because the choice of partition can affect performance [8], we utilized five distinct partitions, which were utilized in every reported experiment.

B. Feature Extraction

Rather than providing raw pixel values to the LSTM network, we instead extracted features from each video, thus significantly reducing the input dimension. For this purpose, we followed one of the approaches used in [2], which was described earlier in [13][14]. Here, the Harris operator [9] is extended to include the temporal dimension, which allows it to find *space-time interest points* – local regions in video where significant spatiotemporal variations occur. For each such region, a *histogram of optical flow* (HOF) descriptor is computed; each descriptor is a real-valued vector of 90 elements. These descriptors are then provided to the network one at a time, in temporal order (for descriptors extracted from the same video frame, the order is arbitrary). An existing implementation was utilized to perform feature extraction (URL: http://www.di.ens.fr/~laptev/download.html; version 1.1 was used); the software was configured with default parameters.

We note that HOF descriptors do not provide an optimal feature set; as discussed in Section II.B, it was shown that LSTM achieves better performance when the features are also computed by a neural network (such as a convolutional network or an auto-encoder). However, our present focus is upon evaluating the relative robustness of the LSTM under various experimental conditions (rather than its absolute performance under the best conditions), and so we found HOF descriptors to be sufficient for our purposes.

C. Long Short-Term Memory

For brevity, we give an informal discussion of long short-term memory (a full description is provided in [11]; a more compact mathematical definition is given in [16]). A LSTM architecture consists an input layer, an output layer, and a hidden *memory block* layer. In Fig. 2, we provide a diagram of a single memory block, which consists of four specialized neurons: a *memory cell*, an *input gate*, a *forget gate*, and an *output gate*. The memory cell and the gates each receive a connection from every neuron in the input layer; furthermore, outgoing connections extend from the memory cell (but not the gates) to every neuron in the output layer. The memory cell further has a recurrent connection to itself (allowing its value to persist through time); gates determine the extent to which (a) the current input influences the value of the memory cell (input gate), (b) the cell's old value is lost (forget gate), and (c) the cell's new value propagates to the output layer (output gate). Through gated control (which is postulated to exist in the human brain [7]), the network can effectively maintain and make use of past observations. There are also *peephole* connections, which allow a gate to know the true state of the memory cell, before it is modulated by the output gate.

In our LSTM networks, there were 90 input neurons (one for each element in the HOF descriptor), 50 memory blocks (each with a memory cell and an input, forget and output gate), and 6 output neurons (one for each action class). During a single time step, the input neurons are activated with values of some HOF descriptor. Subsequently, memory cells and gates compute activation values based on the inputs and on previous memory cell values. Finally,

activations propagate to the output layer, and the process is then repeated for the next HOF descriptor in the sequence. Each output neuron applies the softmax activation function to its weighted input; this ensures that the sum of all output activations is equal to 1, and allows for a probabilistic interpretation of the outputs (i.e., for each action class, the network produces a degree of belief that the observed action belongs to that class). When evaluating the network's performance, we aggregated the network's decisions over a given test sequence; more precisely, for each output neuron, we computed the sum of activation values produced for every descriptor in the sequence; subsequently, the neuron with the highest sum was said to correspond to the class predicted by the network.

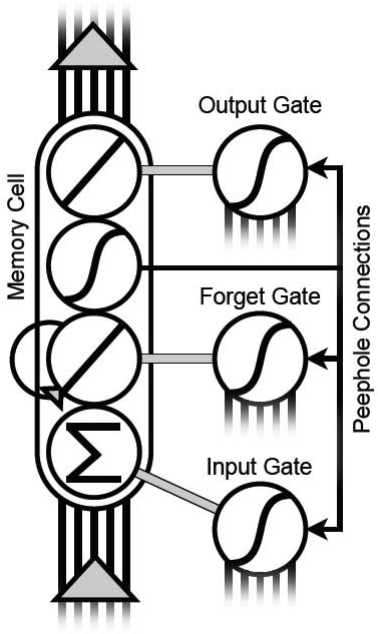

Fig. 2. A cross-section of a LSTM network, with a single memory block, and connections from the input layer (bottom) and to the output layer (top). Multiple solid lines denote full connections between layers; single solid lines indicate pairwise connections; grey bars represent the modulation of the memory cell via gates. The figure was obtained from [16].

In each of our experiments, a network's initial weights were drawn randomly from a uniform distribution U(-0.1, 0.1). The networks were then trained via online backpropagation, minimizing the cross-entropy error function [10], with a learning rate of 0.0001, for 2000 epochs. During a given epoch, training examples (sequences of HOF descriptors) were presented in a randomized order. The correct output vector (with a value of 1 for the output neuron corresponding to the action class, and values of 0 for the other output neurons) was presented to the network after each HOF descriptor.

We note that our methodology is somewhat similar to what was utilized in [2], but with several differences. For example, in [2], momentum was used during training, and each memory cell contained recurrent connections to all other memory cells. We have found that the absence of such connections was somewhat beneficial to performance, as we describe in the next section.

IV. Results

A. Baseline Performance

To establish a baseline for subsequent experiments, we applied our methodology with a full training set size (16 subjects), maximum decision time (the entire input sequence is presented to the network) and non-degraded (original) video quality. As described in Section III.A, networks were trained and evaluated over five independent trials, where each trial involved a different partitioning of the data into the training and test set. As reported in [2], an additional validation set was not necessary, because overtraining was not observed; once trained, the networks' performance did not change significantly, with additional epochs.

In TABLE I, we present the confusion matrix C for the networks after 2000 epochs of training, averaged over the five trials. The rows of the matrix represent true action labels, whereas the columns correspond to labels output by the network; an entry C(i, j) corresponds to the percentage of videos with action i (in the test set) that were classified (by the network) as j; empty cells correspond to values of 0. For each action j, we compute the Type I / false positive error (i.e., the action is erroneously reported) as $\sum_i C(i, j) - C(j, j)$; these are listed at the bottom of the matrix. Similarly, the Type II / false negative error (i.e., the network fails to report the action) of action i is expressed as $\sum_j C(i, j) - C(i, i)$, and shown in the rightmost column. Not surprisingly (and consistent with literature, e.g., [14]), the greatest degree of confusion exists between the "Jogging" and "Running" actions, whereas the remaining actions are distinguished with a high degree of effectiveness.

TABLE I
THE AVERAGE CONFUSION MATRIX (WITH TYPE I AND TYPE II ERRORS) FOR LSTM'S PERFORMANCE ON THE KTH DATASET, IN THE BASELINE CASE

Output / **True**	Boxing	Clapping	Waving	Jogging	Running	Walking	**Type II**
Boxing	**95.9**	2.0	2.0			0.1	**4.1**
Clapping	2.5	**94.4**	3.0			0.1	**5.6**
Waving	1.3	3.1	**95.7**				**4.3**
Jogging	0.1	0.1		**81.7**	13.9	4.2	**18.3**
Running				20.4	**78.5**	1.1	**21.5**
Walking				1.3	0.3	**98.5**	**1.5**
Type I	**3.9**	**5.2**	**4.9**	**21.7**	**14.2**	**5.6**	

The average accuracy measure (proportion of correct classifications, over all videos in the test set) for the five trials is 90.7%, with a sample standard deviation of 0.9%. As we show in TABLE II, the figure is somewhat higher than what was reported in [2], which also applied LSTM to

HOF descriptors (second row), in one of the reported experiments. Because we likely utilized different partitions of the KTH dataset into the training and test sets, we cannot conclusively state that our improvement is due to differences in the network architecture or training regime, as opposed to partition choice (where the latter has been shown to affect performance [8]). However, to establish a *possible* cause for the difference, we ran our five trials with the *same* partitions that we utilized earlier, but with full recurrent connections between LSTM memory cells, which were also utilized in [2]. This yielded a classification accuracy of 85.3% (as reported in the third row of TABLE II), and a sample standard deviation of 2.5%. This suggests that, all other parameters kept constant, the effect of such recurrent connections is detrimental, and *may* partially explain the performance difference. Finally, we mention that our result is slightly below the best reported performance (to our knowledge) achieved on the given dataset and feature combination (KTH and HOF), as provided in the fifth row of the table. However, in that study (which also likely utilized different training set / test set partitions), the HOF descriptors were first clustered into "visual words", and each video was represented as a histogram of such words; these "bag of words" histograms were then classified by the SVM. By contrast, we fed the HOF descriptors into the LSTM network directly as a sequence, without additional preprocessing. The similarity in results suggests that the LSTM is well-adapted for automatically forming appropriate representations based on input features.

TABLE II
PERFORMANCE OF SEVERAL TECHNIQUES ON THE KTH DATASET WITH HOF DESCRIPTORS AS INPUT FEATURES

Approach	Study	Result
LSTM without recurrent cell connections	This	**90.7%**
LSTM with recurrent cell connections	[2]	87.8%
LSTM with recurrent cell connections	This	85.3%
SVM with "bag of words"/clustering	[21]	92.1%

Briefly, we also report the overhead associated with our approach. On a desktop machine with a 2.80 GHz 4 core Intel® Xeon® CPU and 4 GB of RAM, running Windows 7 and Java version 1.7 (the networks were implemented in Java), training a single network (for 2000 epochs) required in the order of one day of computational time. However, once trained, classification is very fast: in one second, the network can process HOF features extracted from over 17,000 frames of video (feature extraction itself is much slower, at approximately 12 to 15 frames per second; each frame contained an average of 1.77 features). Finally, when serialized to disk, each network required approximately 177 kB of storage.

B. *Effects of Training Set Size*

Given our baseline performance, we now begin to perturb the experimental conditions by considering the effects of training set size. Specifically, for each of the five partitions (trials) utilized in the previous experiment, we reduced the training set exponentially (from 16 to 8, 4, 2 and finally, 1 subject), while maintaining the same 9 subject test set. We then trained and evaluated a network for each partition.

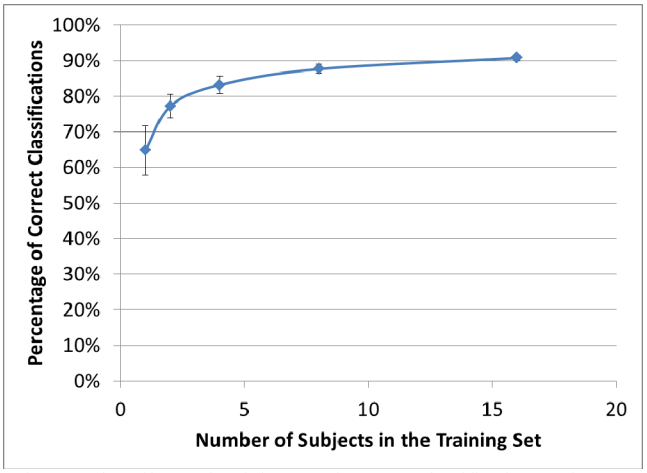

Fig. 3. The effects of training set size upon classification performance. Error bars represent the sample standard deviation (over the five trials) in this figure, and in all other figures where they appear.

Fig. 3 displays the average case performance (over five trials) for each training set size. The approximate number of training video sequences per action can be computed as the number of subjects (shown on the horizontal axis) multiplied by 16, since each subject typically performs a given action in four different scenarios, and for each scenario, the video is usually split into four subsequences, as discussed in Section III.A; the actual number depends on the specific partition. The curve shows that size reductions do not have a very strong impact on performance, until only one subject remains in the training set; here, performance is significantly reduced, and its fluctuation (between partitions) is greatest, as indicated by the error bars. This may suggest that by observing actions performed by more than one distinct subject, the network can better generalize to recognizing actions from previously unseen subjects, but that very large training sets are not necessary for good performance.

As we discussed in Section II.C, an existing study evaluated the performance of the support vector machine (SVM) and sparse representation-based classification (SRC) with respect to training set size, on the KTH dataset [22]. The study utilized different increments for varying training set size (typically, the set contained around 6, 12 or 18 subjects); however, by interpolation, we infer that LSTM achieves performance that is comparable to that of the SRC, and slightly better than that of the SVM. We note, however, that a somewhat richer set of input features was utilized in the earlier study (containing histograms of oriented gradients in addition to histograms of optical flow).

C. *Effects of Input Sequence Size*

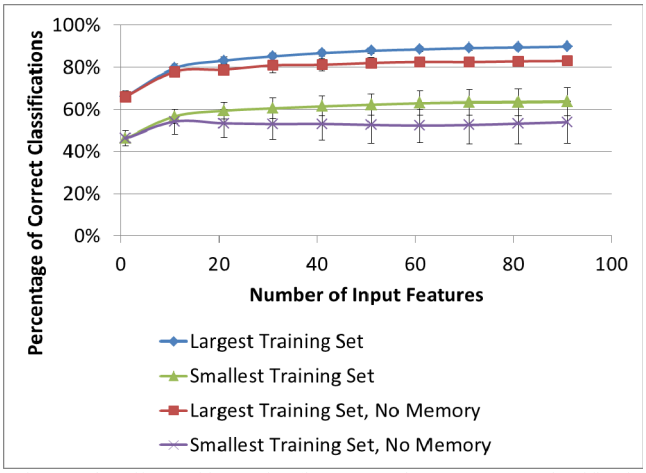

Fig. 4. The effects of input size (latency) and memory upon classification performance.

Since LSTM networks are able to perform sequence processing, they need not accept all available input data prior to making a decision. In our next set of experiments, we sought to determine the extent to which the networks' performance varies with the amount of input data provided (alternatively, we can interpret this as the amount of time available to the network for decision making, or the latency). Specifically, we applied the networks trained with data from 16 subjects and 1 subject (the extreme cases in Fig. 3), and measured their performance (without retraining) on different-sized prefixes of input sequences. In other words, for every video sequence in the test set, only the first k HOF descriptors were provided to the network, where k varied from 1 to 91, in increments of 10.

In Fig. 4, we illustrate the networks' average performance (over five trials/networks) for different values k. The curves labeled "Largest Training Set" (16 subjects) and "Smallest Training Set" (1 subject) suggest that while performance does decrease as k is reduced, the degradation is relatively graceful. Even for just a single HOF descriptor, the networks' average classification accuracy is 66.1% and 46.2%, when trained on data from 16 subjects and 1 subject, respectively; for 20 descriptors, the corresponding figures are 83.0% and 59.4%. As mentioned earlier, approximately 1.77 descriptors were extracted from an average frame; thus, at 25 frames per second, 20 descriptors correspond to less than half a second of video, in the average case. We postulate that performance could be further improved with a richer set of features; for example, in [17], it was found that good classification performance can be achieved on the KTH dataset by extracting local shape descriptors (in addition to optical flow descriptors) from just a few frames.

In interpreting these results, we must recall (from Section III.C) that a network's decision is computed by aggregating each of its 6 output activations over the entire test sequence of k inputs, and selecting the maximum total activation. Thus, an important question arises: if we increase the size of the input sequence, does decision accuracy improve simply as a result of this aggregation (where the overall decision will be correct so long as most per-input decisions are correct), or does memory play a role, as we expect? To answer this question, we evaluated our networks as before, but prior to providing each input, a given network's memory activations were reset. In essence, the network had to classify each HOF descriptor without reference to any previous observations. The resulting performance is shown by the "Largest Training Set, No Memory" and "Smallest Training Set, No Memory" curves in Fig. 4. The figure illustrates that when multiple inputs are given to the network, memory indeed provides a benefit, by allowing new inputs to be interpreted within the context of those observed in the past. In the following section, we demonstrate that this benefit increases under certain types of video quality degradation.

D. *Effects of Video Quality*

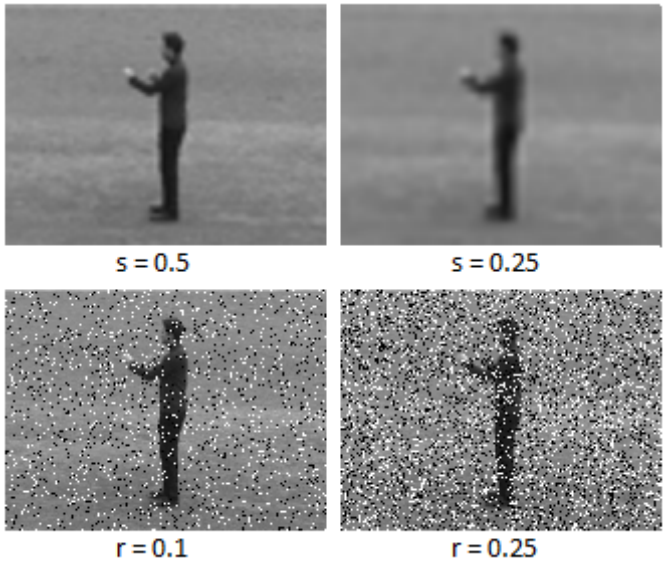

Fig. 5. The effects of degradation on a single video frame (the original, non-degraded frame is shown on the left side of Fig. 1). Versions with reduced resolution are provided at the top of the figure; noisy versions are depicted at the bottom. The effects of dropped frames are not shown, since any kept frame remains unaffected; for p = 0.5, the video appears slightly "choppy"; for p = 0.75, the visual effect is quite severe.

In our final set of experiments, we evaluated the sensitivity of our approach to several forms of video quality degradation. In particular, we examined the impact of:

- *Dropped frames* (which can result in conditions of poor network connectivity). Here, with an independent probability p, any given frame (except the first) is replaced with the (identical) previous frame. The length of the video is thus preserved.
- *Noise* (which can occur due to unfavorable lighting and weather conditions). Here, a ratio r of the pixels in a given frame is made black or white (this results in "salt and pepper" noise).
- *Reduced resolution* (which may arise because of camera limitations, or as a result of bandwidth-conserving measures). Here, each video frame is scaled down by a factor s, and subsequently, scaled up by the same factor

(with anti-aliasing). This preserves the size of each video frame, but causes information/resolution loss.

Using MATLAB's Image Processing Toolbox, we created 6 modified versions of the KTH dataset, three with mild (p = 0.5, r = 0.1, s = 0.5) forms of degradation (each version only had one degradation type applied to it), and three with severe (p = 0.75, r = 0.25, s = 0.25) degradation levels. Here, the terms "mild" and "severe" are somewhat subjective, and indicate the relative visual impact of a given degradation, which is illustrated in Fig. 5. As before, for each of the six degraded datasets, we extracted HOF descriptors, and trained/evaluated LSTM networks.

Resulting classification performance is given in Fig. 6; once again, 16-subject and 1-subject training sets were considered, and each data point is an average over five training/test set partitions (trials). The figure shows that performance is essentially unaffected by a loss in resolution, whereas the introduction of noise and (more so) the removal of frames have a more pronounced effect. Nonetheless, the LSTM network exhibits relatively graceful degradation with respect to these perturbations. In a study mentioned in Section II.C, where a SVM was utilized for classification, performance deteriorated much more significantly with respect to increasing noise and decreasing frame rate; reductions in resolution were only considered in combination with rotation, and this also led to poor performance [18]. However, we note that in the reported experiments, the SVM was trained on data from the original set of videos (with high quality), and then tested on data from the same videos, but with various types of quality degradation. In our experiments, we followed the opposite approach: the training set and the test set consist of distinct videos, but both sets have the same degradation operation applied. Thus, while we can claim that LSTM is relatively robust to the forms of degradation considered herein, it may be important to include low-quality videos in the training set, in order to achieve good performance.

Finally, as before, we evaluated our networks on the degraded test sets, but with memory reset before each input; the results are provided by the "No Memory" curves in Fig. 6. Under severe noise and frame drop rates, memory was found to be particularly beneficial. We can postulate that as the quality of input features (HOF descriptors) decreases due to these forms of deterioration, it becomes more and more difficult to classify each feature individually, without considering features that were observed in the past.

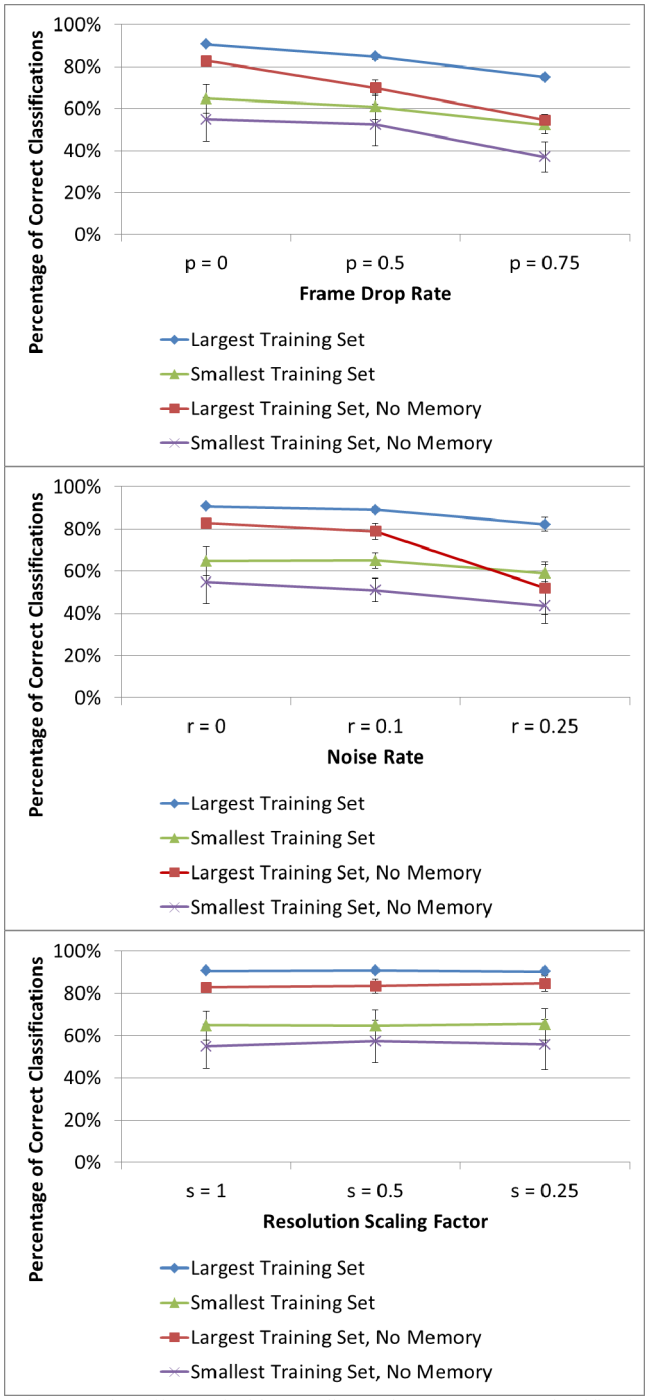

Fig. 6. The effects of video quality and memory upon classification performance.

V. Discussion, Conclusions and Future Work

In real-world applications, it is expensive to produce large quantities of labeled training data; available input sequences may be limited by tight decision making deadlines, and video quality may be affected by sensor limitations, data link constraints and environmental conditions. In this paper, we have evaluated the robustness of the LSTM neural network's performance on the human action recognition problem, with respect to these complications. We have shown that the network's classification accuracy degrades relatively gracefully under the effects of reduced training set sizes, shortened video sequence durations and poor video quality. These results suggest that LSTM has potential for performing action recognition in practical scenarios.

In order to provide a possible explanation for LSTM's robust performance, we observe that it effectively compresses the input sequence by mapping this sequence to its memory cells, through gated control. This dimensionality reduction process is tightly integrated with the classification process; i.e., through a unified training procedure, a LSTM network learns not only how to classify sequences, but also, what information in a sequence will most effectively support classification, and we postulate that this is essential to robust processing. Notably, information requirements can change

as conditions degrade: for example, Fig. 6 shows that as noise or frame loss is introduced, it becomes increasingly more important to process each input in the context of past inputs. By contrast, most existing machine learning techniques (e.g., a typical SVM) require that a long, variable-length sequence (e.g., of HOF descriptors) is first transformed into a compact input (e.g., a histogram of visual words), so that it can be classified in one shot. This separate preprocessing step may not be sufficiently aware of what information is truly important for classification, in a given set of circumstances. Although techniques have been developed for the direct processing of sequences [6], some (e.g., hidden Markov models) have difficulty capturing relationships between inputs that are non-local in time, while others (e.g., conditional random fields) can suffer from high computational overhead. LSTM is emerging as a potentially powerful alternative to such methods, as it is able to tractably extract and correlate temporally distributed information.

Future work can explore ways to further improve robustness. For example, we observed that for the lowest training set size considered, performance is considerably reduced; in order to effectively handle such (or even smaller) training sets, it may be beneficial to incorporate unlabeled data (which is much easier to obtain) into the training process; for example, the network may be trained to both auto-encode sequences (this does not require labels) and to classify them. Further benefits can be gained from utilizing more sophisticated feature extraction techniques, including neurocomputational preprocessors [2][3][4]. Evaluation should be performed on progressively more complex datasets, and the effects of other types of degradation (e.g., errors in the training labels [11][14], other factors that affect video quality [18], etc.) should be studied. Of particular relevance is the effect of object occlusions upon performance, where one object is hidden from view by another. These can arise in videos with multiple interacting objects (people, vehicles, etc.), and present a particular challenge in *activity recognition*, where an activity consists of multiple actions that may be distributed in time and/or space. We postulate that the memory control mechanisms of LSTM will enable it to effectively process sequences where a long lag occurs between actions, and would provide even greater benefit than what we observed for simple actions (in this study). Furthermore, a recent generalization of the LSTM training algorithm (LSTM-g) [16] is capable of training much more sophisticated LSTM-like architectures (e.g., with multiple memory block layers), which may potentially achieve improved performance in such complex activity recognition scenarios.

ACKNOWLEDGMENT

We thank Moez Baccouche (of CNRS/INSA de Lyon), Lee Whitt (of Northrop Grumman) and Peter Chen, Mun Wai Lee, Renato Levy, Vikram Manikonda, David Mihalcik, Goutam Satapathy and Thomas Wavering (of Intelligent Automation, Inc.) for their support in this research.

REFERENCES

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia and A. Baskurt, Action Classification in Soccer Videos with Long Short-Term Memory Recurrent Neural Networks, *20th International Conference on Artificial Neural Networks*, 2010.

[2] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia and A. Baskurt, Sequential Deep Learning for Human Action Recognition, *2nd International Workshop on Human Behavior Understanding*, pp. 29-39, 2011.

[3] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia and A. Baskurt, Sparse Shift-Invariant Representation of Local 2D Patterns and Sequence Learning for Human Action Recognition, *21st International Conference on Pattern Recognition*, 2012.

[4] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia and A. Baskurt, Spatio-Temporal Convolutional Sparse Auto-Encoder for Sequence Classification, *British Machine Vision Conference*, 2012.

[5] N. Beringer, A. Graves and J. Schmidhuber, *Classifying Unprompted Speech by Retraining LSTM Nets*, Technical Report No. IDSIA-07-05, Dalle Molle Institute for Artificial Intelligence, 2005.

[6] T. Dietterich, Machine Learning for Sequential Data: A Review, *Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pp. 15-30, 2002.

[7] M. Frank, B. Loughry and R. O'Reilly, Interactions Between the Frontal Cortex and Basal Ganglia in Working Memory, *Cognitive, Affective and Behavioral Neuroscience* 1:137-160, 2001.

[8] Z. Gao, M.-y. Chen, A. Hauptmann and A. Cai, Comparing Evaluation Protocols on the KTH Dataset, *First International Workshop on Human Behavior Understanding*, pp. 88–100, 2010.

[9] C. Harris and M. Stephens, A Combined Corner and Edge Detector, *Alvey Vision Conference*, pp. 147–152, 1988.

[10] G. Hinton, Connectionist Learning Procedures, *Artificial Intelligence* 40:185–234, 1989.

[11] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, *Neural Computation* 9:1735-1780, 1997.

[12] M. Jordan, Attractor Dynamics and Parallelism in a Connectionist Sequential Machine, *8th Annual Conference of the Cognitive Science Society*, pp. 531-546, 1986.

[13] I. Laptev, On Space-Time Interest Points, *International Journal of Computer Vision* 64(2/3):107-123, 2005.

[14] I. Laptev, M. Marszałek, C. Schmid and B. Rozenfeld, Learning Realistic Human Actions from Movies, *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[15] S. Mahmoud, *Identification and Prediction of Abnormal Behaviour Activities of Daily Living in Intelligent Environments*, Dissertation, Nottingham Trent University, 2012.

[16] D. Monner and J. Reggia, Generalized LSTM-like Training Algorithm for Second Order Recurrent Neural Networks, *Neural Networks* 25(1):70-83, 2012.

[17] K. Schindler and L. van Gool, Action Snippets: How Many Frames Does Human Action Recognition Require?, *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[18] J. Stöttinger, B. Goras, T. Pönitz, N. Sebe, A. Hanbury and T. Gevers, Systematic Evaluation of Spatio-Temporal Features on Comparative Video Challenges, *10th Asian Conference on Computer Vision*, 2010.

[19] J. Stöttinger, S. Zambanini and R. Khan, FeEval – A Dataset for Evaluation of Spatio-Temporal Local Features, *20th International Conference on Pattern Recognition*, pp. 499-503, 2010.

[20] P. Turaga, R. Chellappa, V. S. Subrahmanian and O. Udrea, Machine Recognition of Human Activities: A Survey, *IEEE Transactions on Circuits and Systems for Video Technology* 18(11):1473-1488, 2008.

[21] H. Wang, M. Ullah, Kläser, I. Laptev and C. Schmid, Evaluation of Local Spatio-Temporal Features for Action Recognition, *British Machine Vision Conference*, 2009.

[22] Z. Zhang, *Vision-based Human Action Recognition: A Sparse Representation Perspective*, Thesis, University of Nebraska – Lincoln, 2012.