Continuous Gesture Recognition through Selective Temporal Fusion

Pradyumna Narayana
Department of Computer Science
Colorado State University
Fort Collins, CO, USA
prady@cs.colostate.edu

J. Ross Beveridge
Department of Computer Science
Colorado State University
Fort Collins, CO, USA
ross@cs.colostate.edu

Bruce A. Draper
Department of Computer Science
Colorado State University
Fort Collins, CO, USA
draper@cs.colostate.edu

***Abstract*—Gesture recognition is an important task with the potential to revolutionize human/computer interfaces (HCI). Gestures, however, are dynamic. While a few gestures may be static poses, most gestures are complex temporal sequences of motions. For most HCI applications, gestures must be recognized in real-time in streaming data. Therefore, most recognition systems analyze each frame as it comes in, fusing data across time to detect gestures. This paper presents results of the first systematic study of temporal fusion techniques for streaming gesture recognition. These results show that the choice of the best fusion strategy depends on whether the input is global (i.e. full-frame) or a spatially focused window, and on whether the input is unprocessed RGB or depth depth versus a flow field.**

This conclusion is then used to extend a state-of-the-art architecture for isolated gesture recognition, FOANet [24], to continuous gesture recognition. The result is a system that established a new state-of-the-art for recognition performance on the ChaLearn ConGD data set, with a mean Jaccard Index of 0.77 compared to the previous best result of 0.61. This paper also establishes a baseline of performance for the newer, continuous version of the NVIDIA dataset.

I. Introduction

Gestures are a natural form of human communication. In conversations among people, they convey information about the intentions, interests, feelings and ideas of a speaker [16]. As artificial intelligence becomes more sophisticated there is wide-spread interest in gesture-driven interfaces [33], [30] and in combined speech and gesture interfaces[28], [12], [26], [41], [19], [32]. Before gestures can be used for HCI, however, computers must learn to recognize gestures in continuous streams of RGBD data.

One of the challenges in gesture recognition is integrating information over time. Many gestures are more than just static poses; they are temporal sequences of motions. Information therefore needs to be integrated over time. In addition, continuous data streams contain series of distinct gestures, not just single gestures. Good gesture recognition systems fuse information within gestures but not across the (unknown) temporal boundaries between gestures.

Most gesture recognition systems extract features or labels individually from each frame, and then fuse the information across time. This paper presents the first systematic study of temporal information fusion in such systems. It is based on the idea that a prototypical "channel" in a gesture recognition system processes the input stream either one frame at a time, or as a small sliding window of temporally adjacent frames. A 2D convolutional neural network (CNN) then processes this input; in the experiments in this paper, we use ResNet [11]. The CNN can be used to produce one of two outputs: a *feature vector*, by which we mean the output of the last convolutional layer, or a *label vector*, which is the result of passing the feature vector through a fully connected layer and a softmax layer. These outputs are then smoothed or fused over time to predict a label for each frame, since continuous data streams contain sequences of gestures.

Three methods of temporal information fusion are compared: *late pooling* estimates a label vector for every frame and then smooths the label vectors over time. *Feature pooling* extracts feature vectors rather than label vectors from every frame, smooths the feature vectors, and the trains a fully connected layer and softmax layer to predict labels given the smoothed feature vectors. Finally, *recurrent neural networks* are trained to fuse either feature vectors or label vectors over time. Long Short-Term Memory (LSTM) networks are the most common recurrent neural nets used for this purpose, and are the recurrent nets tested in this paper.

As discussed below, most current gesture recognition systems are multi-channel, meaning that they use multiple CNN channels to process different versions of the input stream. Systems may assign channels to input modalities (e.g. RGB, depth, or flow fields), spatial attention windows, or in the case of FOANet [24] the cross-product of modalities and attention windows. This paper therefore evaluates information fusion techniques in the context of channels, looking for interactions between fusion techniques and input types.

Several interesting conclusion emerge from this study. The first not surprising result is that temporal segmentation fundamentally changes the information fusion problem. Many gesture recognition challenges, including the recent ChaLearn IsoGD competition[36], focus on short video clips containing one gesture each. In essence, they temporally segment the gesture stream. When gesture are segmented there is no predictable difference between the temporal fusion techniques; they all work equally well [23]. In continuous streams with multiple gestures, however, this isn't true. In continuous streams, LSTMs outperform late pooling and feature pooling

overall. Interestingly, however, there is a strong interaction between channel type and the best information fusion technique for continuous data. Global channels, meaning channels that are not focused on a spatial attention window, do best with feature pooling. Spatially focused RGB and depth channels do best late pooling. Spatially focused flow field channels perform best with LSTMs. Thus, although LSTMs are the best information fusion technique on average, still better results can be achieved by matching the information fusion technique to the channel type.

We put these conclusions to the test in the context of FOANet and the 2017 ChaLearn continuous ConGD dataset. Matching the information fusion technique to the channel type generates state-of-the-art recognition results, beating the previous results published in [36]. We also tested our conclusion in the context of the continuous version of the NVIDIA gesture data set [21]. Since no previous results have been published on this data we cannot do a comparison to other work, but we show that matching the information fusion technique to the channel type produces better results than using any of the three information fusion techniques alone.

No empirical comparison of techniques can include all possible variations. The biggest omission in this study is that it does not include 3D convolutions as a temporal information fusion technique. 3D convolution systems do not process frames independently, and therefore do not face the information fusion challenge studied here. However, 3D convolution techniques are not appropriate for streaming applications since the entire video is processed as a unit. This paper can therefore be viewed as comparison of streaming infusion fusion techniques. Another omission is that this paper only considers information fusion within channels, and does not address how information is fused across channels.

The main contributions of this paper therefore are:

1) A systematic study of 3 temporal information fusion strategies in gesture recognition, showing that
    a) There is a strong interaction between channel type and the best information fusion technique for continuous data.
    b) LSTMs are the best temporal fusion technique overall.
    c) The best performance is achieved by matching the information fusion strategy to the channel type.
2) State-of-the-art recognition accuracy on the ChaLearn ConGD [37] data set.
3) The first reported results on the continuous version of the NVIDIA [21] data set, again showing the benefits of matching the fusion strategy to the channel type.

The rest of the paper is organized as follows: Section II reviews related work on continuous gesture recognition, including FOANet. Section III describes the methodology used to compare temporal fusion strategies and to look for interactions between input types and temporal fusion strategies. Experimental results are provided in section IV for the ChaLearn ConGD data set and Section V for the NVIDIA data set. Section VI concludes the paper.

II. Related Work

Gesture recognition is a large and rapidly-evolving field of study. Older gesture recognition surveys describe a range of techniques [33], but recent work is dominated by deep learning approaches, as described by Asadi-Aghbolaghi *et al*.[1]. Of particular importance in that survey is the two-stream architecture of Simonyan and Zisserman [35]. They introduced the use of parallel processing channels for different modalities in the context of action recognition. This idea has since been widely adopted, to the extent that all the entries in the 2017 ChaLearn IsoGD competition used multi-channel architectures [36], as does the more recent FOANet [24]. Insights and results pertaining to multi-channel architectures are therefore of broad interest to the gesture recognition community [25].

Karpathy *et al*. [15] introduced a two-channel architecture where the channels were differentiated by spatial attention rather than modality. They had two channels, one that processed the full frame at a low resolution and one that processed the middle of the frame at a higher resolution, under the assumption that the target of interest would be centered. More recently, Wang *et al*. [38] dedicated channels to spatial attention windows based on hand tracking. Narayana *et al*.also dedicated channels to process head, right hand and left hand regions [26] for real-time gesture recognition in cooperative tasks [40]. FOANet [24] combines the modality and spatial approaches by assigning a processing channel to every combination of modality and spatial attention window.

This paper looks at temporal information fusion within multi-channel architectures for gesture recognition. Since the experiments are performed on the ChaLearn ConGD and NVIDIA datasets, we briefly review the state-of-the-art with regard to these data sets. The ChaLearn ConGD dataset is a multimodal (RGB-D) dataset with 47,933 gesture instances in 22,535 videos. It has 249 gestures labels performed by 21 different individuals. Each video consists of one or more gestures, and the goal is to recognize gestures within continuous videos. The dataset is split into three standardized and mutually exclusive subsets: training, validation, and test.

Both of the top two performers in the 2017 ConGD challenge [36] segmented the videos into gestures and then classified the resulting segments. Liu *et al*.segmented the videos based on the data-set-specific observation that subjects raise their hands at the beginning of gestures and put them down again at the end. They used a two-channel 3D convolutional network to generate features from the RGB and depth modalities respectively without having to do temporal information fusion, and trained an SVM to determine the final label for each segment [20]. This turned out to be the winning strategy.

Wang *et al*.achieved second place by segmenting videos into gesture frames and transition frames using a two-class CNN [44], [39], [43]. The middle frame of a continuous

Fig. 1. Different gestures in one video of ChaLearn ConGD Dataset [37].

segment of transitional frames is considered to be a boundary, thereby segmenting the video into gestures. The depth segments are converted into Motion Selective Images and classified with a CNN. The RGB segments and the saliency maps are processed by CNNs and the resulting features are passed to an LSTM that labels the segment. The softmax scores from the saliency maps, depth and RGB channels are then averaged together.

Camgoz *et al*.used a probabilistic forced alignment approach to jointly segment and recognize gestures [5]. They train a 3D CNN to classify gestures (plus a silence class to identify neutral gestures) by extracting windows from a prior distribution of likely regions. The trained 3D CNN is used to calculate the posterior distribution which is used as the prior to sample windows for the next training stage. This process is repeated until the recognition performance of the network has converged on validation set. During inference, they run the 3D CNN on all windows and segment the video at frames where the silence class probability is maximum. The softmax scores in individual segments is averaged to assign a label to the segment.

Unlike the ChaLearn ConGD dataset, the NVIDIA Dynamic Hand Gesture Dataset is targeted to a specific domain: gestural interfaces for cars [21]. This dataset provides RGB and depth videos of 25 gestures performed by 20 subjects recorded with the SoftKinetic DS325 sensor. The gestures are performed in a car simulator under both bright and dim lighting. Molchanov *et al*.report the best results on this dataset using recurrent 3D convolutional networks. The NVIDIA dataset has videos that start with users in a neutral position (hands on the steering wheel). They perform a gesture and then return to the neutral position. These videos can be considered continuous videos with three gestures: *drive, gesture, drive*. Molchanov *et al*.used this dataset to report the accuracy on isolated gesture recognition (with the *drive* portions of the videos removed), but not on continuous recognition. This paper reports the first continuous gesture recognition results for this dataset.

This paper extends FOANet to process continuous data sets. FOANet uses 12 parallel channels to label gestures: one for each combination of 4 modalities and 3 attention windows. As described in [24], FOANet processed temporally segmented videos by averaging CNN softmax vectors over time. The temporal information fusion work in this paper came out of the desire to extend FOANet to continuous domains such as ConGD and NVIDIA, and the result was a system establishing a new state-of-the-art for performance on these two data sets.

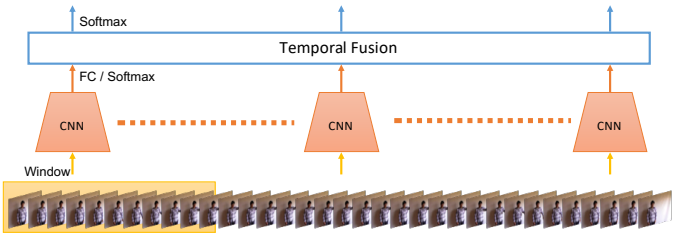

Fig. 2. Architecture of prototypical information channel. The input to the channel is a sliding window of 10 frames which is processed by by CNNs (orange). At every time step the CNNs analyze 9 frames it has seen before and one new frame. The FC features and/or softmax scores from CNNs are temporally fused resulting in a label at every timestep.

A variety of temporal fusion approaches has been proposed as an effective way to integrate information over time, such as rank pooling [7], [42], CNN-based temporal pooling [15], [45], [3], [18], [2]. All of the above approaches are proposed for video classification task and they temporally pool information from the entire video. So, they are not appropriate for streaming applications where every frame must be labeled with minimal latency. As this paper extends FOANet for continuous gesture recognition problem, we only compare the temporal fusion methods appropriate for this task.

III. Methodology

Our approach to studying temporal information fusion is to begin with a prototypical information channel. We then vary (1) the source of the input and (2) the temporal fusion strategy. The goal is to find the best temporal fusion strategy, and to determine if the best strategy is a function of the input source.

The input to a prototypical information channel is a sliding window of ten frames, as shown in Figure 2. As a result, at every time step the channel analyzes 9 frames it has seen before and one new frame. This is consistent with both Simonyan and Zisserman's two-channel architecture [35] and the current state-of-the-art, FOANet [24]. The window of ten frames is processed by a convolutional neural net with a ResNet architecture. The temporal fusion strategy is tasked with fusing the output of the network over time to produce a gesture label for each frame of input.

A. Temporal Fusion

We investigate three temporal fusion mechanisms: *Late Pooling*, which combines information at the label (softmax) level, *Feature Pooling*, which smooths features prior to the

fully connected layer, and recurrent networks in the form of *LSTMs*. This section gives more information about these three techniques, but it is also worth noting that other less successful techniques were evaluated and rejected, including Max Pooling of features over small temporal windows, GRU recurrent networks [4], stacked LSTMs [10] and nested LSTMs [22]

1) Late Pooling:: At every time step, channels take a 10 frame window centered on the current frame as input and produce a vector of softmax scores. Late Pooling fuses information across time by convolving this sequence of softmax vectors with a 1-D Gaussian kernel. More formally, for every frame t, a channel produces a vector of C softmax scores, where C is the number of classes. These vectors can be stacked together to form a matrix S of dimensions $C \times T$. Let G be the 1-D Gaussian kernel of length 5σ+1, calculated as $G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, where σ is the standard deviation and μ $(= 2.5\sigma + 1)$ is the mean of the Gaussian kernel. The pooled softmax scores S_{lp} are calculated as $S_{lp} = S * G$, and the argmax of S_{lp} at time t gives the predicted gesture label at time t. The value of σ is set as $\frac{1}{4}$th of the average gesture length in the dataset, so that 2 standard deviations of the Gaussian kernel covers the average length of the gesture.

2) Feature Pooling:: Whereas Late Pooling fuses labels, Feature Pooling fuses feature vector (FC) - the output of the last convolutional layer. These features encode information from a 10 frame sliding window. Similar to Late Pooling, they are convolved with a 1-D Gaussian kernel. A fully connected layer is then trained to classify the pooled features, producing a logit vector that is fed into a softmax function. More formally, for every frame t, a channel produces a feature vector FC of length $|FC|$ (2048 for global nets and 2062 for focus nets). These vectors can be stacked together to form a matrix FC_T of dimensions $|FC| \times T$. Let G be the 1-D Gaussian kernel of length 5σ+1, calculated as $G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, where σ is the standard deviation and μ $(= 2.5\sigma + 1)$ is the mean of the Gaussian kernel. The pooled features FC_{Pool} are calculated as $FC_{Pool} = FC_T * G$. A fully connected layer is then trained to classify these FC_{Pool} features.

3) LSTMs:: Unlike Feature Pooling and Late Pooling, recurrent neural networks explicitly model sequences of variable length and take temporal orderings into account. We model RNNs as LSTM networks [13], [8] applied to both the feature vectors of the CNNs (as in Feature Pooling) and the label vectors (as in Late Pooling), as shown in Figure 3. The LSTM outputs are passed through a softmax classifier to label every sliding window. Based on pilot experiments, we use a 1024 memory cell LSTM.

B. Channel Inputs

In order to look for interactions between types of source data and the best information fusion strategy, we follow the lead of FOANet and consider 12 types of input for information channels. Every channel receives data with a specific modality and target. There are four data modalities: data may be RGB, depth, or a flow field computed from RGB or depth data.

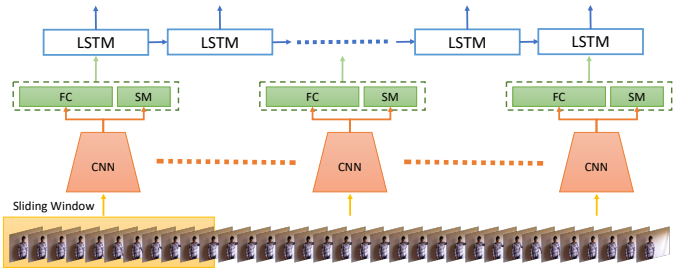

Fig. 3. Architecture of Recurrent Neural Networks. A sliding window of 10 frames is processed by CNNs (orange) and the FC features and softmax scores from CNNs are stacked together (green). These stacked features are processed forward through time by LSTMs. A softmax layer predicts the gestures at each time step.

There are also three targets: a channel may process the whole frame (so-called global channels), or it may process just an attention window around the left hand or right hand. The result is 12 information channels, each corresponding to a unique combination of modality and target.

C. Model Selection

At the heart of every information channel is a convolutional neural network trained with standard deep learning algorithms and techniques. One of these techniques is model selection. CNNs are trained iteratively. One typically observes that the training accuracy increases steadily over time, while the validation accuracy plateaus and then eventually decreases. The standard way to select the best model is to choose the network where validation accuracy is maximum [9].

However, the model that maximizes the validation accuracy is not the best model for all temporal fusion strategies. The maximum validation accuracy model is best for Late Pooling, but not for Feature Pooling or LSTMs. Our experiments suggest that earlier models selected when the validation accuracy first starts to plateau are less prone to overfitting and are better for Feature Pooling and LSTMs. This model can be found by fitting a line to the last $\frac{1}{4}$th of validation accuracies by using least squares approximation and taking the first model to cross this line, as shown in Figure 4.

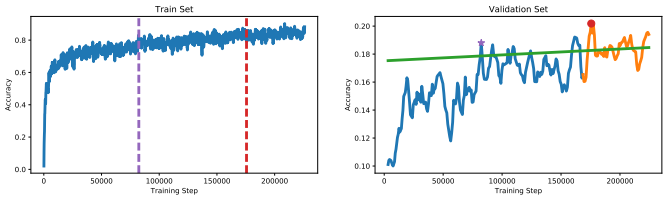

Fig. 4. Model selection for temporal fusion. The left frame shows shows training accuracy as function of training steps for one CNN, while the right frame shows validation accuracy as a function of training steps. Late Fusion is optimized by selecting the model that maximizes validation accuracy, shown as a red dot in the right frame. Gaussian Pooling and LSTMs are optimized by less overfit models produced earlier in the training. For these techniques, we fit a line (shown in green) to the last quarter of the validation accuracy points (shown in orange). The first model that crosses the green line is selected.

D. Implementation Details

Spatial attention requires that the left and right hands be detected and tracked to create focused channels. Faster R-CNN [34], [14] is used to detect right and left hands for ConGD dataset. The NVIDIA dataset is simpler in this respect; the right hand is always the closest object to camera. (Left hands do not appear in this dataset.) As in [24], optical flows are computed from adjacent frames using pyflow [29].

IV. ChaLearn ConGD Experiments

This section compares the performance of the three temporal fusion strategies on all 12 information channels, using data from ChaLearn ConGD data set. The results are unexpected, and suggest that different fusion strategies are best for different types of information channels. To measure the impact on an integrated system, we then extend FOANet to use the best temporal fusion strategies within each channel, producing a new state-of-the-art level of performance for the ConGD data set.

A. Training Process

To accelerate training, we "warm start" the weights of the focus channel nets from ResNet-50 [11] trained on ImageNet [6], and then use the weights of the trained right hand nets to fine-tune the global channels. To warm start focus channels from ResNet-50, the first pretrained convolutional layer weights $(7\times7\times3\times64)$ are repeated 10 times and stacked together $(7 \times 7 \times 30 \times 64)$ to account for the different number of input channels (3 vs 30) between ImageNet and focus channels. The fully connected layer weights are randomly initialized and the nets are trained end to end using mini-batch stochastic gradient descent with momentum (0.9) and a random batch of size 64. The input volume is randomly cropped to $100 \times 100 \times 30$ and flipping is performed so that a single sample can help train both right hand and left hand nets. The learning rate is initially set to $2e-4$ and decays exponentially with a decay factor of 0.7 and decay steps of 40,000.

The fine-tuned right hand focus channels are used as a warm start for the respective global channels. For example, the RGB global channel is trained by fine-tuning the RGB right hand focus channel. The global channel nets are also trained end to end using mini-batch stochastic gradient descent with the same momentum term, batch size and learning rate rules as focus channel nets. The input volume is randomly cropped to a $224 \times 224 \times 30$ volume and random flipping is performed for data augmentation.

When training temporal fusion methods, the FC features and softmax scores from all channels are precomputed. Feature pooling is trained by first convolving the FC features with a Gaussian kernel and then training a fully connected layer weights using the Adam optimizer [17] with a batch size of 64. The initial learning rate is set to 0.01 for first 10,000 steps, and is decreased to 0.001 till 20,000 steps and is further decreased to 0.0001. Training stops after 100,000 iterations. LSTMs are trained by stacking the FC features and softmax scores as input and training LSTM weights using the Adam optimizer [17] with a batch size of 64 and a dropout of 0.25. The cost function is calculated over all the frames of input and the gradients are backpropagated at each frame. The learning rate rule and training iterations is similar to Feature Pooling. Sparse network fusion weights are learned by precomputing the softmax scores of all the channels and convolving these scores with a Gaussian kernel of size 51. The weights are trained using the Adam optimizer [17] with a batch size of 32. The learning rate rule and training iterations are similar to Feature Pooling.

B. Inference Process

During inference, data is passed through the convolutional networks without augmentation (cropping or flipping). Features and softmax scores are calculated at every timestep. The FC features from global channels are convolved with a Gaussian kernel and passed through the fully connected layer trained by Gaussian Pooling. The FC features and softmax scores from focused flow field channels are stacked together and passed through the LSTM. The softmax scores from all channels are convolved with a Gaussian kernel (Late Pooling) and the scores are stacked together and multiplied by the fusion layer weights and the diagonal of the resulting matrix is extracted. These softmax scores are further fused using Late Pooling and the argmax of the softmax scores is the predicted gesture label for the timestep.

C. Results

Table I records the accuracy of every information channel using each of the three temporal fusion strategies. Results are reported independently on the validation and test data, as in standard in the ChaLearn challenge. Each row corresponds to one of the 12 processing channels. The values in a row are mean Jaccard Index scores.

Two quick observations can be made based on Table I. The first is that no channel by itself is state-of-the-art. Fusion across channels is important. This is not surprising; the previous state-of-the-art algorithms used multiple channels as well. The second is that when performance is averaged across all 12 channels, LSTMs are a better temporal fusion mechanism than Late Pooling or Feature Pooling. But of course there is no reason to use the same fusion mechanism in all channels.

The most significant finding in Table I lies in the pattern of maximum values, highlighted with light blue backgrounds. For all four global channels, for both the validation and test data sets, the highest mean Jaccard Index is achieved using Feature Pooling. Focused channels behave differently, however. Focused channels of "raw" data, i.e. channels that process windows of RGB or depth values, achieve the highest mean Jaccard Index using Late Pooling on both the validation and test sets. Focused channels of flow fields achieve their highest values with LSTMs.

This is a strong pattern, even though some of the numeric differences are small. We have three types of channels: global, focused raw and focused flow field. We have four channels of each type, and each channel is tested on two data sets

TABLE I

MEAN JACCARD INDEX SCORES OF CHANNELS FUSED BY DIFFERENT TEMPORAL FUSION MECHANISMS ON VALIDATION AND TEST SET OF CHALEARN CONGD. THE NUMBERS IN BOLD WITH BLUE BACKGROUNDS REPRESENT THE BEST SCORES FOR A GIVEN CHANNEL.

	Valid			Test		
Temporal Fusion / Channels	Late Pooling	Gaussian Pooling	LSTM	Late Pooling	Gaussian Pooling	LSTM
RGB Global	0.2210	**0.2501**	0.2320	0.2323	**0.2379**	0.2164
Depth Global	0.2784	**0.3286**	0.3084	0.3221	**0.3423**	0.3353
RGB Flow Global	0.396	**0.4408**	0.3983	0.4012	**0.4188**	0.3735
Depth Flow Global	0.308	**0.3476**	0.2980	0.3387	**0.3684**	0.3204
RGB Left	**0.2337**	0.2042	0.2207	**0.2162**	0.1822	0.2126
RGB Right	**0.2745**	0.2482	0.2455	**0.3954**	0.3455	0.3557
Depth Left	**0.3104**	0.2865	0.3018	**0.2836**	0.2704	0.2831
Depth Right	**0.4225**	0.3690	0.3948	**0.5293**	0.4754	0.5178
RGB Flow Left	0.3336	0.3225	**0.3383**	0.3038	0.2735	**0.3041**
RGB Flow Right	0.4439	0.4235	**0.4522**	0.4912	0.4709	**0.4918**
Depth Flow Left	0.2715	0.2893	**0.3104**	0.2728	0.2643	**0.2753**
Depth Flow Right	0.3693	0.3896	**0.4194**	0.5055	0.4632	**0.5130**

TABLE II

MEAN JACCARD INDEX SCORES OF TEMPORAL FUSION ON THE VALIDATION AND TEST SETS OF CHALEARN CONGD. SELECTIVE FUSION COMBINES THE BEST TEMPORAL FUSION STRATEGIES FOR EACH TYPE OF CHANNEL: FEATURE POOLING FOR GLOBAL CHANNELS, LATE POOLING FOR RAW FOCUSED CHANNELS, AND LSTMS FOR FOCUSED FLOW FIELD CHANNELS.

	Valid	Test
Late Pooling	0.7420	0.7481
Gaussian Pooling	0.7387	0.7036
LSTM	0.7650	0.7615
Selective Fusion	**0.7791**	**0.7740**

(validation and test). We therefore have 8 3-way comparisons among fusion mechanisms for each channel type, for a total of 24 3-way comparisons. Across all of these comparisons, every channel type has a single fusion mechanism that is always best. This best mechanisms depends on the channel type. The data set is also very large, so none of the differences can be attributed to small sample sizes. When comparing temporal fusion techniques, all 3 algorithms are applied to the same samples and produce answers that are either correct or incorrect. We can therefore test for statistical significance using McNemar's test. In all 24 cases, the maximum value is significantly greater than the other two at a level of $p = 1e - 05$.

D. Cross-channel Results

Table II shows the results of fusing information across all 12 channels. Once the data has been fused temporally within each channel, cross-channel fusion is performed using a sparse fusion network, as in [24]. The Late Pooling, Feature Pooling and LSTM columns show the result of using a single temporal fusion mechanism across all channels. The "selective" column shows the result of using Feature Pooling on global channels, Late Pooling on raw focused channels, and LSTMs on focused flow field channels. Not surprisingly given the results in Table I, the best result is achieved by selecting the fusion mechanism based on the channel type.

The selective row in Table II corresponds to FOANet extended to process continuous data streams by adding temporal fusion, where the fusion strategy is determined by the channel type. Table III compares this "selective FOANet" to previously published results on the ChaLearn ConGD data set. It shows a significant increase in performance over all prior systems. Interestingly, by combining the results of the two tables we see that any temporal fusion strategy is sufficient to beat the previous best algorithms, but that the selective strategy outperforms any single fusion strategy.

TABLE III

CHALEARN CONGD 2017 RESULTS. ENTRIES ARE ORDERED BY THEIR PERFORMANCE ON TEST DATA. RESULTS ON SYSTEMS OTHER THAN OURS WERE PREVIOUSLY REPORTED IN [36].

System	Valid	Test
Selective FOANet	**0.7791**	**0.7740**
ICT_NHCI [20]	0.5163	0.6103
AMRL [39]	0.5957	0.5950
PaFiFA [5]	0.3646	0.3744
Deepgesture [31]	0.3190	0.3164

E. Analysis

Table I reveals a clear relationship between channel type and the optimal temporal fusion strategy, but it doesn't explain why. Since the architectures of the channels are the same, there must be some difference in the data being processed that makes one fusion model more or less appropriate than another. We have looked at many properties of the FC features and softmax vectors being produced by the CNNs inside the information channels. Most do not correlate to the best choice of temporal fusion strategy. Only one property seems to predict the choice of temporal fusion strategy: *prediction stability*. We define prediction stability as the frequency with which a CNN predicts the same label (whether right or wrong) on two consecutive time steps. Figure 5 shows the prediction stability of each of the 12 channels, with global channels shown in blue to the left, raw focused channels shown in orange in the middle, and focused flow field channels shown in purple on the right.

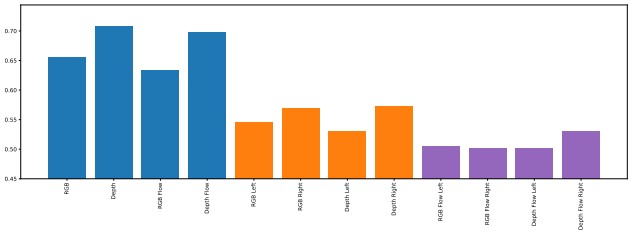

Fig. 5. Bar plots showing prediction stability of CNNs for all 12 channels. Prediction stability is defined as the frequency with which a CNN predicts the same label on two consecutive time steps. Global channels show the most prediction stability, while raw focused channels show intermediate stability and focused flow field channels are the least stable.

Figure 5 suggests that the CNNs within individual channels are not as stable as one might guess. Even the global depth channel, which is the most stable, only predicts the same label for two consecutive time steps about 75% of the time. Other channels are as low as 50%. Some of this can be explained by the large label set (the likelihood of picking the

TABLE IV
MEAN JACCARD INDEX SCORES OF CHANNELS ON THE NVIDIA DATASET FOR DIFFERENT TEMPORAL FUSION STRATEGIES. THE LAST COLUMN SHOWS THE RESULT OF FUSING ALL 8 CHANNELS TOGETHER. THE LAST ROW SHOWS THE COMBINED SCORES THAT RESULT FROM FUSING ALL 12 CHANNELS WITH TEMPORAL FUSION STRATEGIES SELECTED BASED ON CHANNEL TYPE.

	Global				Focus				Sparse Fusion
	RGB	Depth	RGB Flow	Depth Flow	RGB	Depth	RGB Flow	Depth Flow	
Late Pooling	0.4632	0.4836	0.5549	0.4895	**0.4792**	**0.5265**	0.5154	0.4924	0.7176
Gaussian Pooling	**0.5402**	**0.5532**	**0.5701**	**0.5364**	0.4137	0.5048	0.5071	0.4314	0.6989
LSTM	0.5047	0.5326	0.5618	0.5304	0.4537	0.5092	**0.5621**	**0.5149**	0.7428
Selective FOANet	**0.5402**	**0.5532**	**0.5701**	**0.5364**	**0.4792**	**0.5265**	**0.5621**	**0.5149**	0.7615

same label twice at random is $\frac{1}{249}$) and the large number of frames between gestures during which no label is obvious. Nonetheless, the lack of stability reinforces the importance of temporal fusion to smooth out data over time.

More importantly for this analysis, prediction stability correlates to the choice of temporal fusion strategy. Global channels are the most stable, and they perform best with Feature Pooling. Raw focused channels have intermediate stability and perform best with Late Pooling. Focused flow field channels are the least stable, and perform best with LSTMs.

We can only speculate as to why prediction stability correlates to the relative performance of temporal fusion strategies. LSTMs model long-term dependencies, which may allow them to detect long-term patterns in highly noisy data. Ng *et al*.also noticed that LSTMs outperform pooling-based strategies when processing optical flow fields [27] and suggested that LSTMs process "optical flow in a manner which lends itself to late model fusion" [27], and that this is not possible for pooling based methods.

Feature Pooling, on the other hand, is designed to filter white noise. Global channels are the most stable channels, as can be seen in Figure 5. We hypothesize that their errors are better modeled as temporal white noise, so that Feature Pooling performs best on global channels. We cannot prove this conjecture, however, nor can we explain why Late Pooling outperforms other methods on raw focused channels.

V. NVIDIA EXPERIMENTS

The pattern in Table I was so striking that we wanted to make sure it would hold for a different data set. The videos in the NVIDIA dataset are quite different from ChaLearn. All gestures are performed with the right hand, and the left hand is never in the field of view. Therefore there are only 8 (instead of 12) FOANet channels. Every subjects starts in a neutral position (hands on steering wheel), performs a gesture, and then returns to the neutral position. Every video therefore has three labels: drive, gesture, and drive. Unfortunately, the timing is too regular: subjects always start gesturing around frame 130 ($\pm$20) and the gestures last for 80 frames. To make the data more realistic, we select random start and end points for each video in the dataset such that all videos remain at least 60 frames long.

Table IV shows the results on the NVIDIA dataset. The CNNs, temporal fusion methods and sparse fusion network are trained in the same way as on ChaLearn ConGD (see Section IV-A), except that random flipping is not performed and the focus channels are only trained on right hands. Table IV displays the same pattern as Table I, albeit with fewer channels. Once again, the global channels perform best with Gaussian Pooling, the (now 2) raw focused channels perform best with Late Pooling, and focused flow field channels perform best with LSTMs. As a side benefit, we can now also report the first performance result for the continuous NVIDIA data set, with cross-channel fusion producing an overall Jaccard Index of 0.7615.

VI. CONCLUSION AND FUTURE WORK

Continuous gesture recognition is more challenging than isolated gesture recognition. In isolated gesture recognition, almost any temporal information fusion technique will work, because all the frames of the video are part of the same gesture. Even simple voting or averaging strategies work reasonably well [24], [23]. Continuous videos, on the other hand, contain transitions between gestures that make temporal information fusion harder. This paper looks at three information fusion strategies that can be applied to continuous data: late pooling, feature pooling, and LSTMs. Experiments show that the best fusion strategy depends on the type of input. Global channels (i.e. channels that process the whole frame) perform best with feature fusion, while spatially focused RGB and depth channels perform best with late pooling. Spatially focused flow field channels perform best with LSTMs. This pattern holds across two domains (ConGD and NVIDIA) and a total of 20 channels. When FOANet is modified to fuse data with this approach, a new state-of-the-art for performance is established on both data sets.

The FOANet architecture is highly specific: global channels process the whole video and look for gross motions, while focused channels detect and process each hand. The main disadvantage of this method is that the spatial information is lost by the time the channels are fused. We anticipate that performance may be improved by fusing channels earlier. In particular, we would like to explore an early fusion strategy that is inspired by foveal and peripheral vision in the human eye, where the hands have high resolution (similar to foveal vision) while other parts of the image are processed at lower resolution (similar to peripheral vision).

REFERENCES

[1] M. Asadi-Aghbolaghi, A. Clapes, M. Bellantonio, H. J. Escalante, V. Ponce-López, X. Baró, I. Guyon, S. Kasaei, and S. Escalera. A survey on deep learning based approaches for action and gesture recognition in image sequences. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 476–483. IEEE, 2017.

[2] A. Cherian and S. Gould. Second-order temporal pooling for action recognition. *International Journal of Computer Vision*, 127(4):340–362, 2019.

[3] A. Cherian, P. Koniusz, and S. Gould. Higher-order pooling of cnn features via kernel linearization for action recognition. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 130–138. IEEE, 2017.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[5] N. Cihan Camgoz, S. Hadfield, and R. Bowden. Particle filter based probabilistic forced alignment for continuous gesture recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[7] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5378–5387, 2015.

[8] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[9] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[10] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

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

[12] S. G. Hill, D. Barber, and A. W. Evans III. Achieving the vision of effective soldier-robot teaming: Recent work in multimodal communication. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, pages 177–178. ACM, 2015.

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

[14] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE CVPR*, 2017.

[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[16] A. Kendon. *Gesture: Visible Action as Utterance*. Cambridge University Press, 2004.

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

[18] P. Koniusz, A. Cherian, and F. Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *European Conference on Computer Vision*, pages 37–53. Springer, 2016.

[19] N. Krishnaswamy, P. Narayana, I. Wang, K. Rim, R. Bangar, D. Patil, G. Mulay, R. Beveridge, J. Ruiz, B. Draper, et al. Communicating and acting: Understanding gesture in simulation semantics. In *IWCS 201712th International Conference on Computational SemanticsShort papers*, 2017.

[20] Z. Liu, X. Chai, Z. Liu, and X. Chen. Continuous gesture recognition with hand-oriented spatiotemporal feature. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[21] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016.

[22] J. R. A. Moniz and D. Krueger. Nested lstms. *arXiv preprint arXiv:1801.10308*, 2018.

[23] P. Narayana. *Improving Gesture Recognition through Spatial Focus of Attention*. PhD thesis, Colorado State University, 8 2018.

[24] P. Narayana, R. Beveridge, and B. Draper. Gesture recognition: Focus on the hands. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] P. Narayana, R. Beveridge, and B. Draper. Analyzing multi-channel networks for gesture recognition. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019.

[26] P. Narayana, N. Krishnaswamy, I. Wang, R. Bangar, D. Patil, G. Mulay, K. Rim, R. Beveridge, J. Ruiz, J. Pustejovsky, and B. Draper. Cooperating with avatars through gesture, speech and action. In *Proceedings of the IEEE Intelligent Systems Conference (IntelliSys)*, 2018.

[27] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4694–4702. IEEE, 2015.

[28] T. Oka and K. Matsushima. Multimodal manipulator control interface using speech and multi-touch gesture recognition. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, pages 21–22. ACM, 2015.

[29] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] A. Pereira, J. P. Wachs, K. Park, and D. Rempel. A user-developed 3-d hand gesture set for human–computer interaction. *Human factors*, 57(4):607–621, 2015.

[31] L. Pigou, M. Van Herreweghe, and J. Dambre. Gesture and sign language recognition with temporal residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3086–3093, 2017.

[32] J. Pustejovsky, N. Krishnaswamy, B. Draper, P. Narayana, and R. Bangar. Creating common ground through multimodal simulations. In *Proceedings of the IWCS workshop on Foundations of Situated and Multimodal Communication*, 2017.

[33] S. S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, 2015.

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[35] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[36] J. Wan, S. Escalera, A. Gholamreza, H. J. Escalante, X. Baró, I. Guyon, M. Madadi, A. Juri, G. Jelena, L. Chi, et al. Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges. In *ChaLearn LaP, Action, Gesture, and Emotion Recognition Workshop and Competitions: Large Scale Multimodal Gesture Recognition and Real versus Fake expressed emotions, ICCV*, volume 4, 2017.

[37] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and S. Z. Li. Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 56–64, 2016.

[38] H. Wang, P. Wang, Z. Song, and W. Li. Large-scale multimodal gesture recognition using heterogeneous networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3129–3137, 2017.

[39] H. Wang, P. Wang, Z. Song, and W. Li. Large-scale multimodal gesture segmentation and recognition based on convolutional neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[40] I. Wang, M. B. Fraj, P. Narayana, D. Patil, G. Mulay, R. Bangar, J. R. Beveridge, B. A. Draper, and J. Ruiz. Eggnog: A continuous, multi-modal data set of naturally occurring gestures with ground truth labels. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 414–421. IEEE, 2017.

[41] I. Wang, P. Narayana, D. Patil, G. Mulay, R. Bangar, B. Draper, R. Beveridge, and J. Ruiz. Exploring the use of gesture in collaborative tasks. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 2990–2997. ACM, 2017.

[42] J. Wang, A. Cherian, and F. Porikli. Ordered pooling of optical flow sequences for action recognition. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 168–176. IEEE, 2017.

[43] P. Wang, W. Li, Z. Gao, C. Tang, and P. O. Ogunbona. Depth pooling based large-scale 3-d action recognition with convolutional neural networks. *IEEE Transactions on Multimedia*, 20(5):1051–1061, 2018.

[44] P. Wang, W. Li, S. Liu, Y. Zhang, Z. Gao, and P. Ogunbona. Large-scale continuous gesture recognition using convolutional neural networks. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 13–18. IEEE, 2016.

[45] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.