Gesture Recognition: Focus on the Hands

Pradyumna Narayana, J. Ross Beveridge, Bruce A. Draper
Colorado State University
{prady, ross, draper} @cs.colostate.edu

Abstract

Gestures are a common form of human communication and important for human computer interfaces (HCI). Recent approaches to gesture recognition use deep learning methods, including multi-channel methods. We show that when spatial channels are focused on the hands, gesture recognition improves significantly, particularly when the channels are fused using a sparse network. Using this technique, we improve performance on the ChaLearn IsoGD dataset from a previous best of 67.71% to 82.07%, and on the NVIDIA dataset from 83.8% to 91.28%.

1. Introduction

Gestures are a natural form of human communication. When accompanying speech, gestures convey information about the intentions, interests, feelings and ideas of the speaker [17]. Gestures are even more important in noisy environments, at a distance, and for people with hearing impairments. In these scenarios, gestures replace speech as the primary means of communication, becoming both more common and more structured [21].

Automatic gesture recognition is therefore an important domain of computer vision research, with applications in Human/Computer interfaces (HCI). Not surprisingly, a large literature has developed on gesture recognition; see [4, 25, 12, 1] for surveys. A good way to measure progress in this crowded field is to look at the ChaLearn challenges, which started in 2011 and have continued through 2017 [11, 8, 10, 9, 7, 6]. The current ChaLearn IsoGD [30] dataset is one of the largest and most varied gesture datasets available, with 249 gestures from a variety of domains including mudras (Hindu/Buddhist hand gestures), Chinese numbers, and diving signals. The ChaLearn 2017 challenge attracted competitors from across the world [29], and the results of that challenge can be reasonably interpreted as reflecting the current state of the art.

If there is a downside to the ChaLearn challenge and the IsoGD dataset, it is that they are not closely tied to any specific HCI application. For this reason, we also track progress on the NVIDIA driving gesture dataset [23], which mimics touch-less interfaces in cars. As shown in Figure 1, the NVIDIA setting is always the same, and the gestures are made by drivers exclusively with their right hands. The NVIDIA dataset is therefore a more focused counterpoint to the wide-open IsoGD dataset.

Figure 1. Example images from the ChaLearn IsoGD dataset (left) and NVIDIA dataset (right). NVIDIA gestures are constrained driving gestures, while IsoGD contains many types of gestures (mudras, diving gestures, etc.) in unconstrained settings.

This paper presents the best results reported so far on the IsoGD and NVIDIA datasets. These results are generated by reintroducing an old idea: focus of attention. Gestures have both global and local components. Some involve sweeping motions of the arms and torso, while others are defined by detailed hand poses. Nonetheless, previous techniques for the ChaLearn and NVIDIA datasets process whole images. In contrast, we train multiple nets with specific purposes: global channels to process the whole video and look for gross motions, and focused channels to detect and process each hand. By fusing information from these channels, we raise the state-of-the-art (SOA) for recognition accuracy from 67.71% to 82.07% for IsoGD, and from 83.8% to 91.28% for NVIDIA.

Our architecture, which we call FOANet, builds on previous systems that use multiple channels to process different data modalities, e.g. [22, 31, 34, 23]. Unlike previous systems, however, FOANet uses spatial focus of attention (FOA) to restrict some channels to focus on specific body parts, namely hands. FOANet introduces a separate channel for every focus region (global, right hand, left hand) and modality (RGB, depth, and two types of flow fields). The

result is 12 channels processing different types of localized data, as shown in Figure 2.

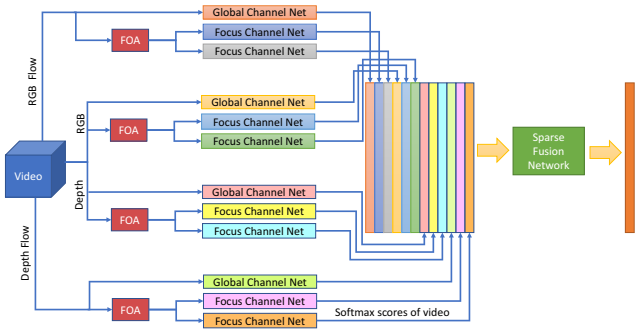

Figure 2. The FOANet Network Architecture. The architecture consists of a separate channel for every focus region (global, left hand, right hand) and modality (RGB, depth, RGB flow and depth flow). FOA module is used to detect hands. The video level softmax scores from 12 channels are stacked together. Sparse fusion combines softmax scores according to the gesture type.

The 12 channels are only useful if the data can be fused back together. It is tempting to train a neural net to fuse the 2,048-dimensional features vectors from all 12 channels, but with only 35K training videos in IsoGD (and far fewer in NVIDIA) there is not enough data to avoid overfitting. This is why many multi-channel systems simply average the channel outputs, e.g. [31, 34, 23]. Instead, FOANet uses a sparsely connected neural layer with one weight per label $\times$ channel. For every gesture type, the sparse fusion layer learns the relative importance of the different spatial regions and data modalities.

In summary, the main contributions of the paper are:

1. State of the art recognition accuracies on the ChaLearn IsoGD [30] and NVIDIA [23] data sets.

2. A novel architecture with focus of attention channels.

3. A novel sparse network architecture for fusing channels.

The rest of the paper is organized as follows: Section 2 reviews the related work on gesture recognition, multi-channel networks, and the datasets used in this paper. Section 3 introduces FOANet and provides the implementation details needed to replicate the results. Experimental results on the ChaLearn IsoGD dataset are provided in section 4, and results on the NVIDIA dataset are presented in section 5. Section 6 concludes the paper.

2. Related Work

ChaLearn LAP RGB-D Isolated Gesture Dataset (IsoGD) [30] is a large multi-modal dataset for gesture recognition. The dataset has 249 gesture labels performed by 21 different individuals. It is split into three mutually exclusive subsets: training, validation, and test. The training set consists of 35,878 videos from 17 subjects, the validation set consists of 5,784 videos from 2 subjects, and the test set consists of 6,271 videos from the other 2 subjects.

There have been ChaLearn gesture recognition challenges every year since 2011; the 2017 challenge reports results on IsoGD [29]. Miao *et al*. [22] won the 2017 challenge using a C3D model [28] and Temporal Segment Network [32] to extract features from RGB, depth and flow fields. Features within each modality are fused using canonical correlation analysis, and an SVM labels videos based on the fused features from the different modalities. The SYSU ISEE team processed skeleton data in addition to RGB, depth and flow fields. They used a combination of rank pooling, LSTMs and temporal streams, and fused the streams using average fusion. Other participants of the challenge [31, 34] also used C3Ds and some form of LSTM for temporal fusion. The resulting channels are fused together by averaging softmax scores.

The closest method to ours is the heterogeneous networks of Wang *et al*. [31]. They use two types of networks: 3D ConvLSTMs to recognize gestures in videos and CNNs to recognize gestures from dynamic images constructed by rank pooling. They apply these networks at two spatial levels, namely body and hands. The networks are run on RGB and depth data and scores from the 12 modalities are averaged together. Wang *et al*. detect bounding boxes around the hands in every frame using F-RCNN [26] and eliminate parts of the scene not within the bounding box circumscribed by the hands to avoid overfitting to the background. For gestures involving big motions and/or two hands, the bounding boxes approach the full size of the image, defeating the purpose of the hand channel. The hand level networks of Wang *et al*. are designed to eliminate background but not to focus attention directly on the hands. In contrast, we detect the right and left hands and select attention windows around them, so that our focus nets are always focused on hands alone. Karpathy *et al*. also had a similar idea of training a global and focus net [16]. However, they fix attention to the center of the frame, relying on camera bias. This will not work on ChaLearn and NVIDIA data sets, where subjects are not centered on the frame.

Although ChaLearn is the largest gesture dataset available, the gestures are drawn from multiple domains. Recently, Molchanov *et al*. released the NVIDIA Dynamic Hand Gesture Dataset [23]. This dataset consists of 25 human computer interface gestures, performed by 20 subjects indoors in a car simulator with both bright and dim artificial lighting. The SoftKinetic DS325 sensor is used to acquire front view color and depth videos and a top-mounted DUO 3D sensor is used to record a pair of stereo-IR streams. Sub-

jects perform gestures with their right hand while observing the simulators display and controlling the steering wheel with their left hand. The dataset is split into a training set of 1,050 videos and a test set of 482 videos. Molchanov *et al*.'s recurrent three-dimensional convolutional neural network is the best reported method on this dataset. Similar to some of the entries of ChaLearn challenge, Molchanov *et al*. use a 3D-CNN to extract local spatial-temporal features and a recurrent network to aggregate transitions. Unlike the ChaLearn competitors, Molchanov *et al*. use connectionist temporal classification as the cost function to train the network. RGB, depth, optical flow, IR image and IR disparity streams are fused by averaging the softmax scores.

3. Approach

We propose a new approach to gesture recognition that reintroduces the old idea of spatial focus of attention. Our approach builds on the multi-channel approaches described above, in which different channels process different data modalities. We expand on this idea by dedicating channels to 3 spatial attention regions: one for the whole scene, and one each of the hands. The idea is to create an architecture that reflects the structure of gestures, which are combinations of large body movements and fine hand motions.

Figure 2 shows our proposed architecture. It has three main components: 1) a focus of attention mechanism, 2) 12 separate global and focused channels, and 3) a fusion mechanism. The task of the FOA component is to detect hands. We use Liu *et al*.'s hand detection network [19, 20] on the ChaLearn data set. For the NVIDIA depth data, we use the heuristic that the right hand is the closest object to the sensor, while for NVIDIA RGB images we use the HandSegNet of Zimmerman and Brox [35]. The global and focused channels are CNNs modeled after ResNet [13], except that focused channels have additional structure to process the positions of the attention windows. Finally, fusion occurs through a sparse network that learns which channels are important for each gesture.

The rest of this Section describes our approach in more detail. Section 3.1 describes the global and focus channels. Section 3.2 explains the sparse fusion network that combines information across channels. The FOA mechanisms and other details required to reconstruct the system are explained in Section 3.3.

3.1. Global & Focused Channels

As shown in Figure 2, global channels process the whole video (one channel per data modality), while focused channels process each hand (one channel per hand/modality). Global and focus nets are architecturally similar, with some differences to account for the spatial location of the attention windows within the larger frame.

Global Channels: Global channels are based on 50 layer deep residual networks [13, 14]. ResNet-50 is a high-performing network on the ImageNet challenge [5]. Although there are deeper versions of ResNet (ResNet-101, ResNet-152, ResNet-1001) and better performing architectures on ImageNet like Inception-V4 [27] and Squeeze and Excitation Network [15], ResNet-50 is selected for practical reasons: we need to train many channels, and each ResNet-50 fits on a single GPU in our lab. Unlike the original ResNet that takes a single image as input, the input to a global channel is a stack of images. More precisely, the input is a temporal window of 10 image frames that captures local motion information. Let w and h be the width and height of the video. For an arbitrary frame t, we stack 10 consecutive frames around t (frames between [t-4,t+5]) to form a 30 channel input volume $I_g^{w\times h\times 30}$. The first 4 and last 5 frames of the video are discarded. Other than the first layer, the convolution and pooling layers of the global channel are the same as in ResNet-50, and produce a 2048 dimensional feature vector as shown in figure 3. Also as in ResNet-50, a fully connected layer followed by softmax produces one output per label from the 2,048 feature vector.

Global channels are trained for four modalities: RGB, depth, optical flow fields from RGB and optical flow fields from depth images. Section 3.3 provides more details about optical flow fields.

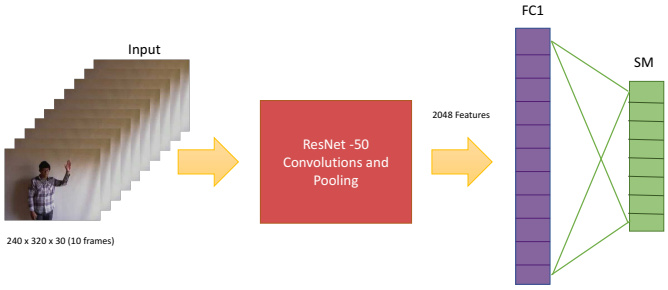

Figure 3. Network Architecture of Global Channels. The input to the network is a stack of 10 images resulting in a $240 \times 320 \times 30$ volume. The input volume is passed through ResNet-50 convolution and pooling layers resulting in 2048 features. A fully connected layer on top produces a vector of softmax scores.

Focus Channels: Similar to the global channels, focus channels take a stack of images as input and use 50 layer deep residual networks [13, 14] as the network architecture. Unlike the global channels, the input image stack is not a stack of whole image, but instead is a stack of spatial image windows focused around one of the hands. For an arbitrary frame t, let (x_1, y_1) and (x_2, y_2) be the top left and bottom right corners of the bounding box centered on a hand. Let $s = max(x_2 - x_1, y_2 - y_1)$ be the maximum side of the bounding box. An input volume$I_f^{s\times s\times 30}$ that is centered on the bounding box is cropped from $I_g^{w\times h\times 30}$. The cropped image stack I_f is then resized to $I_f^{128\times 128\times 30}$ and is given

as input to the focus channels. Section 3.3 provides details about hand detection.

To tell the focus channel where the hands are, we provide 14 additions location features (7 for each hand). The location features are: (x, y) locations of top left and bottom right corners of the bounding box, width and height of the bounding box, and the ratio between the width and height. If only one hand is visible, we set the features of the other hand to zeros. These 14 features are passed through one hidden layer of 14 nodes with a tanh activation function, and the resulting 14 features are appended to ResNet-50's features as shown in figure 4. The resulting feature vector is passed to a fully connected layer for classification.

A separate focus net is trained for each hand. For applications that involve only one hand, as in the NVIDIA data set, a single focus net is trained for each modality. Similar to global nets, focus nets are trained on four modalities: RGB, depth, optical flow from RGB images and optical flow from depth images, resulting in 4 (NVIDIA) or 8 (IsoGD) focus nets depending on the number of hands.

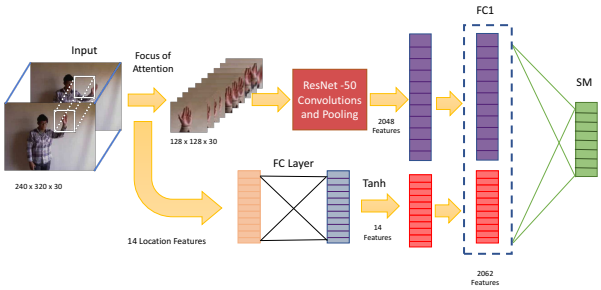

Figure 4. Network Architecture of Focus Channels. The input to the network is a cropped volume centered on hands. The input volume is passed through ResNet-50 convolution and pooling layers. In addition, 14 location features are passed through a fully connected layer of 14 neurons with a tanh non-linearity. These 14 features are concatenated on to the ResNet features, and a fully connected layer on top produces a vector of softmax scores.

Global and focus channels take a 10 frame sliding window as input and produce a vector of softmax scores at each time step. To create a single softmax vector for the whole video, we average the predictions. More formally, for every frame t in video v (excluding the first 4 and last 5 frames), a channel produces a vector of softmax scores of length C, where C is the number of classes. These vectors can be stacked together to form a C x T matrix. The softmax scores for the video v are calculated by taking the mean across the time axis; the argmax of the resulting mean softmax vector gives the gesture class prediction.

3.2. Sparse Network Fusion

The 12 global and focused channels shown in Figure 2 produce 12 response vectors. These vectors need to be combined to produce a single gesture label per video. Many multi-channel systems average response vectors together, as we did for the temporal dimension. This is not the best use of the available information, however; see Section 4.2 below. Unfortunately there is not enough training data to train a fully connected neural layer to fuse the 2,048 or 2,062 dimensional feature vectors produced by the convolutional networks inside each channel. With 12 channels, the concatenated feature vector would be over 24,000 elements long, and the fusion layer as a whole would have to learn over 6 million weights. With only about 35K training videos in the IsoGD dataset, the network would overfit.

We propose a more directed learning mechanism. The goal is to learn the properties of gestures. For example, the diving gestures in ChaLearn were designed to be seen at a distance through murky water, so they involve large arm motions. Mudras, on the other hand, are small dextrous motions of one hand. Our goal is to learn how much weight to assign to a channel, given a gesture, so that global channels are emphasized for diving gestures while right hand channels are dominant for mudras. We therefore fuse channels using a sparsely connected network with one weight per gesture $\times$ channel.

Let n be the number of channels and C be the set of classes. For video v, let $S = [s_1, s_2, s_3, ..., s_n]$ be the softmax scores, where s_i is a vector of length $|C|$. If a hand is never visible or doesn't move throughout the video (average movement less than 4 pixels), the corresponding softmax vector is set to all zeros. For each class $c \in C$, the weight vectors $W_c = [w_{c1}, w_{c2}, w_{c3}, ..., w_{cn}]$ should be calculated to weigh the different channels according to their importance to gesture c.

We pose this problem as a perceptron learning problem where class weights are learned in tandem. Let $W = [W_1, W_2, W_3, ..., W_{|C|}]$ be the weight matrix to be learned. The dimensions of softmax score matrix S are C x n and the dimensions of weight matrix W are n x C. These two matrices can be multiplied to create F: $F = SW$. The dimensions of F are $|C|$ x $|C|$, and the diagonal elements of F represent the softmax scores of classes multiplied with their corresponding class weights ($F_{ii} = S_{i,*}.W*, i$), whereas the off diagonal elements represent softmax scores of a class multiplied with weights of different classes($F_{ij} = S_{i,*}.W_{*,j}$). The off diagonal elements of matrix F are therefore discarded by doing a Hadamard product of F with a $|C| \times |C|$ identity matrix I. A softmax function is applied to the diagonal elements of FI. The weight matrix W is learned by back propagation using cross entropy loss and mini-batch gradient descent. As the off-diagonal elements are zeroed out by the Hadamard product with I, they do not produce derivatives and the weights of a class are effected only by their corresponding softmax scores.

3.3. Implementation Details

This section describes two important components built largely on prior work by others. While not themselves contributions, understanding our system as a whole requires understanding how hands are detected (Section 3.3.1) and how flow fields are computed (Section 3.3.2).

3.3.1 Hand Detection

For ChaLearn, we use the hand detection results provided by Liu *et al*. [19, 20]. They use a two stream Faster R-CNN for hand detection. First depth video is aligned to RGB video and convolutions are run separately on RGB and depth videos. Feature maps from RGB and depth maps are stacked together. A Region proposal network [26] and an object classifier is run on the stacked feature maps. The hand detection results provided by Liu *et al*. do not differentiate between right and left hands. Skeletons extracted from RGB frames using multi-person pose estimation code by Cao *et al*. [3] are used to distinguish left and right hands.

Right and left wrist skeleton estimates are interpolated and extrapolated when necessary to fill in missing skeleton joints in some frames. Then bounding boxes closest to the wrist are found in RGB images using the approach of Liu *et al*. The bounding boxes from RGB images are mapped onto depth images by the following transformation: $D = \frac{R-14}{0.93}$, where R is a coordinate in RGB image and D is its corresponding location in depth image.

For the NVIDIA dataset, hand detection results were not available. However, the hand is the closest object to the camera in the dataset. So, hand can be detected by considering the closest object to the camera in depth videos. To segment hands in RGB videos, we use the HandSegNet of Zimmermann and Brox [35]. HandSegNet is a 16 layer network that is based on and initialized by the person detector of Wei *et al*. [33]. For a given RGB frame, HandSegNet returns a two channel image, one of which is a hand mask and the other one is the background mask.

3.3.2 Optical Flow

Optical flow is computed from two adjacent frames sampled using pyflow [24] - a python wrapper for dense optical flow [2]. As it is computationally not feasible to calculate optical flow on the fly, we pre-compute the flow fields. Moreover, we store the optical flow values as RGB images to make it easy to store and work with the optical flow values. To store the flow fields as RGB images, the horizontal and vertical components of the flow values are clipped at -20, 20. Then magnitude of the both components is calculated. The horizontal, vertical and magnitude components are rescaled to [0, 255] range independently and saved as red, green, blue channels respectively of a RGB image.

4. ChaLearn IsoGD Experiments

To measure the effectiveness of spatial attention channels and gesture-based fusion relative to other techniques, we compare the recognition accuracy of FOANet as shown in Figure 2 to those of previous systems on the ChaLearn IsoGD (this section) and NVIDIA (next section) data sets. Since FOANet significantly outperforms previously published results, we run additional experiments designed to measure the contributions of specific parts of the system.

4.1. Methodology

4.1.1 Experimental Design

The 2017 ChaLearn IsoGD challenge asked participants to classify videos as one of 249 gestures [29]. Participants were given access to a set of 35,878 labeled training videos, and a second set of 5,784 labeled validation videos. Participants were encouraged to develop the best system they could, training on the training videos and testing on the validation videos. At the conclusion of the challenge, participants were given access to a previously sequestered set of 6,271 labeled test videos. They were asked to evaluate their system on the test videos without modification.

Since our system was developed after the challenge deadline, we mimicked this experimental design as closely as possible. We internally sequestered the test videos, and did not test our system on them during development. We incrementally developed our system by training on the training videos and testing on the validation videos. At the end, we evaluated the system only once and without modification on the test videos.

Participants in the challenge generally report two sets of numbers: performance on the validation data, and performance on the test data. In Section 4.2 below, we do the same.

4.1.2 Training Process

The convolutional nets inside the global and focused channels are trained using various forms of "warm starts". The convolutional nets in global channels are fine-tuned from ResNet-50 pretrained on ImageNet [5]. The pretrained ResNet-50 takes 3 channel images as input, whereas our global channel nets takes 30 channels as input (a 10 image stack with 3 bands per image). To account for this, the first pretrained convolutional layer weights $(7 \times 7 \times 3 \times 64)$ are repeated 10 times and stacked together $(7 \times 7 \times 30 \times 64)$. The last fully connected layer weights are randomly initialized and the nets are trained end to end using mini-batch stochastic gradient descent with momentum (set to 0.9) and a random batch of size 64. The input volume is randomly cropped to a $224 \times 224 \times 30$ volume and random flipping is performed for data augmentation. The learning rate lr is ini-

tially set to $2e-4$ and is decayed exponentially with a decay factor df of 0.7 and decay steps ds of 40,000. The decayed learning rate dlr at a step is calculated as $dlr = lr * df^{\frac{step}{ds}}$.

The global channel convnets took 9 days to fine-tune on the ChaLearn dataset using a single Titan X GPU. We used the fine-tuned global channel as a warm start for the respective focus channels. For example, the RGB left hand and RGB right hand focus channels are trained by fine-tuning the RGB global channel. The convolution weights for focus channels are initialized from the pretrained global channels and the fully connected layer weights (location and last fully connected layers) are randomly initialized. The input volume is randomly cropped to a $100 \times 100 \times 30$ volume and random flipping is performed by flipping left hand and using it to train right hand nets and vice versa. Similar to global channel nets, focus 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.

To learn the weights of the fusion layer, the softmax scores of different channels of training data are precomputed. The weights are then trained using the Adam optimizer [18] with a batch size of 32. 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. The training is stopped after 50,000 iterations.

All convolutional networks are trained on training data, and the best model is selected based on its accuracy on the validation set. The best models are then used for testing, and results are reported for both the validation and test set (see Section 4.1.1). All models are trained in Tensorflow on single NVIDIA Titan X GPU and evaluation is done on single NVIDIA GTX 980 GPU.

4.1.3 Inference Process

During inference, data is passed through the convolutional networks without augmentation (cropping or flipping). For global channels, the input volume is $240 \times 320 \times 30$; for focus channels, the input volume is $128 \times 128 \times 30$. For an arbitrary video v and channel c, FC features and softmax scores are calculated at every timestep. These scores are averaged across the video, resulting in a single softmax vector. If a hand is never visible or it's average movement is less than 4 pixels throughout the video, the corresponding softmax scores for that channel are set to all zeros. All the scores are stacked together and are multiplied by the fusion layer weights and the diagonal of the resulting matrix is extracted. The argmax of the diagonal is the predicted gesture label.

4.2. Results

Our method achieves state-of-the-art performance on the ChaLearn IsoGD dataset, as shown in Table 1. Table 1 also shows the top performing entries from the ChaLearn 2017 competition [29]. On the validation data we outperform the previous SOA by 16.5%, with an accuracy of 80.96% compared to the previous best of 64.4%. On the test set we achieve an accuracy of 82.07%, outperforming the previous state-of-the-art by 14.3%.

System	Valid	Test
FOANet (this paper)	**80.96**	**82.07**
Miao *et al.* [22] (ASU)	64.40	67.71
SYSU_IEEE	59.70	67.02
Lostoy	62.02	65.97
Wang *et al.* [31] (AMRL)	60.81	65.59
Zhang *et al.* [34] (XDETVP)	58.00	60.47

Table 1. ChaLearn IsoGD 2017 results. Entries are ordered by their performance on test data. Results on systems other than ours were previously reported in [29].

As already stated, focus of attention and sparse network fusion are the keys to our method. To evaluate the contribution of sparse network fusion, we replace it with average fusion, i.e. averaging the output of the softmax layers of the 12 channels. The average fusion version of FOANet achieves better results than previous methods (67.38% vs 64.40% on validation set and 70.37% vs 67.71% on test set), as shown in Table 2. Therefore, sparse network fusion improves performance by 11.7%.

Another way to interpret this result, however, is that focus of attention channels are surprisingly powerful. The other entries in Table 1 use 3D convolutions and RNNs. Our approach with spatial attention channels outperforms these techniques using only 2D convolutions, averaging across time, and averaging across channels.

To probe further, we applied averaging to all possible subsets of the 12 channels. With averaging as the fusion mechanism, the best performance was achieved by a subset of 7 of the 12 channels: 3 RGB flow channels, 2 depth focus channels, the RGB right hand channel, and the depth flow right hand channel. If we average these 7 channels together, the accuracy is 69.06% on the validation set and 71.93% on the test set, as shown in Table 2. This is roughly 1.5% better than averaging all 12 channels, and suggests that 5 of the channels produce as much noise as information. We see a different pattern with sparse network fusion, however. By using only 7 channels with sparse network fusion, the accuracy decreases to 77.31% on the validation set and 78.9% on the test set. With sparse network fusion the system learns which channels to include for each gesture type, with the result that sparse network fusion benefits from the presence of channels that hurt performance when averaging channels.

We also experimented with training a neural net to fuse the FC feature vectors (2048 for global channels, 2062 for focus channels) from all 12 channels and 7 channels. Unfortunately, this method doesn't perform on par with sparse network fusion or even simply averaging the softmax out-

Fusion	Valid		Test	
	12 Channels	7 Channels	12 Channels	7 Channels
Sparse	**80.96**	**77.31**	**82.07**	**78.90**
Average	**67.38**	**69.06**	**70.37**	**71.93**
Concatenation	56.03	55.29	59.44	58.84

Table 2. Comparison of fusion strategies. Accuracies are shown for FOANet using sparse network fusion, channel averaging, and concatenation for 12 channels (maximal for sparse nets) and 7 channels (optimal for averaging).

Validation Set			
	Global	**Left**	**Right**
RGB	33.22	16.17 (23.41)	41.60 (41.76)
Depth	27.98	23.76 (34.40)	54.91 (55.12)
RGB Flow	46.22	24.14 (34.95)	54.60 (54.81)
Depth Flow	31.66	21.84 (31.62)	48.32 (48.51)
Test Set			
	Global	**Left**	**Right**
RGB	41.27	16.63 (19.55)	47.41 (47.44)
Depth	38.50	24.06 (28.29)	64.44 (64.48)
RGB Flow	50.96	24.02 (28.23)	59.69 (59.73)
Depth Flow	42.02	22.71 (26.70)	58.79 (58.83)

Table 3. Individual channel accuracies on ChaLearn IsoGD validation and test set. The numbers represent the accuracies on all videos of validation and test set. However, not all videos have both hands visible. The accuracies in brackets shows the accuracies on the videos where the particular hand is visible.

puts, as shown in Table 2. The problem is overfitting: there isn't enough triaining data to constrain the weights.

4.3. Analysis of channels

Here we analyze the performance of channels independently and in combination. Table 3 shows the accuracy of each channel on the IsoGD validation and test sets. Unfortunately, the left and right hands are not visible in all videos. Right hands are visible in 5,762 of 5,784 validation videos and 6,267 of 6,271 test videos, or in about 99% of the videos. In contrast, left hands are only visible in 3,994 of 5,784 validation videos and 5,334 of 6,271 test videos, or about 77% of the videos. The numbers in brackets in Table 3 refer to the classification accuracies of focused channels when limited to videos in which the corresponding hand is visible.

A clear pattern emerges in the columns in Table 3: right hands outperforms global channels and global channels outperform left hands in all eight cases. Presuming performance is a guide to where the most useful information resides, the most useful information is in the right hand. This is not surprising since the dataset contains mostly right handed participants, and participants tend to use their left hand only for two-handed gestures. So even when the left hand is visible, it is often idle. However, overall performance is best when all channels are combined, suggesting that the left hand is important for two-handed gestures and that sparse network fusion is able to learn when to pay attention to the left hand.

When we compare the rows in Table 3, the contributions of the different data modalities are more complex. Global channels perform best when they process flow fields extracted from RGB data. This is consistent with the idea that global channels are looking for gross movements. Right hand channels perform best on depth data, suggesting that many of them may be poses rather than motions, although they also perform well on RGB flow fields. Left hand channels perform roughly the same on depth and RGB flow field data. We also note that flow fields extracted from depth data don't perform on par with flow fields extracted from RGB data. This may be attributable to the fact that flow field extraction algorithms are designed for RGB images, not depth images, and suggests an opening for better flow field from depth algorithms.

Next we combine channels from different modalities using sparse network fusion, as shown in Table 4. From the first two fusion columns, we can see that the combination of focus channels is better than the combination of global channels. In fact, the fusion of focus channels is the best combination, short of combining all channels. Moreover, most of the information from focus is contributed by the right hand alone which can be attributed to the right handed bias in the dataset. We also notice that the fusion of RGB and RGB flow nets is better than the fusion of depth and depth flow nets on validation set. However on the test set, depth + depth flow performed better. Looking back at Table 3, we can see that "Depth Right" outperforms all other channels on the test set and that contributed to depth modality's overall performance. Next, we see that the fusion of RGB and depth channels performs on par with the fusion of RGB flow and depth flow channels. We also note that all of the columns in Table 4 except for the global column outperform the previous state-of-the-art.

5. NVIDIA Experiments

5.1. Methodology

5.1.1 Experimental Design

Recently, NVIDIA published a dataset of 25 gesture types intended for touchless interfaces in cars. The dataset consists of 1532 dynamic hand gestures performed by 20 subjects. RGB, depth and a pair of stereo-IR streams are provided for each hand gesture, although we use only RGB and depth streams. The data is split by subject into 1050 training and 482 test videos. As a validation set is not provided with the dataset, we choose 1 subject from the training set to be the validation set. We follow the same experimental design as in ChaLearn by incrementally developing our system by training on the training videos and testing on the validation videos. We evaluated the system only once and without modification on the test videos.

	Validation	Test	Global	Focus	Right	RGB	Depth	Raw	Flow	All
RGB Global	33.22	41.27	✓			✓		✓		✓
RGB Left	23.41	19.55		✓		✓		✓		✓
RGB Right	41.76	47.44		✓	✓	✓		✓		✓
Depth Global	27.98	38.50	✓				✓	✓		✓
Depth Left	34.40	28.29		✓			✓	✓		✓
Depth Right	55.12	64.48		✓	✓		✓	✓		✓
RGB Flow Global	46.22	50.96	✓			✓			✓	✓
RGB Flow Left	34.95	28.23		✓		✓			✓	✓
RGB Flow Right	54.81	59.73		✓	✓	✓			✓	✓
Depth Flow Global	31.66	42.02	✓				✓		✓	✓
Depth Flow Left	31.62	26.70		✓			✓		✓	✓
Depth Flow Right	48.51	58.83		✓	✓		✓		✓	✓
Validation			61.4	**76.76**	**72.64**	**71.41**	**68.56**	**70.69**	**70.49**	**80.96**
Test			67.5	**77.61**	**74.46**	**75.41**	**76.39**	**75.29**	**74.39**	**82.07**

Table 4. Results of fusing different combinations of channels. 'Raw' refers to input from a stack of unprocessed images, whereas 'flow' refers to input of a stack of flow field images. The last column matches the first row of Table 1. Bold-face numbers represent results that are higher than the previous SOA. Note that all combinations involving focus channels beat the previous SOA.

Method	Channels	Accuracy
FOANet	FOA + Sparse Fusion	**91.28**
FOANet	FOA + Avg. Fusion	**85.26**
Human	Color	88.4
Molchanov [23]	All (including IR)	83.8
Molchanov [23]	Depth + Flow	82.4

Table 5. Results on NVIDIA test set. The bold-face numbers represent results that are higher than previously reported results.

	RGB	Depth	RGB Flow	Depth Flow
Global	43.98	66.80	62.66	58.71
Focus	58.09	73.65	77.18	70.12

Table 6. Individual channel accuracies on NVIDIA test set

5.1.2 Training and Inference

The CNNs for the NVIDIA dataset are trained in a similar way to the CNNs for the IsoGD dataset (See Section 4.1.2) with three differences: 1) the CNNs are fine-tuned from the respective channel nets trained on IsoGD; 2) flipping is not used to augment the training set, as people always sit to the left with their left hand on steering wheel and all gestures are performed with the right hand only; 3) only right hand focus channels are trained, since the left hand is never visible. The inference process is similar to the process for ChaLearn as discussed in Section 4.1.3, except that we do not have any still hands in the dataset.

5.2. Results

FOANet performance surpasses both the previous best result and human accuracy, as shown in Table 5. Our method achieves an accuracy of 91.28%, a 7.5% increase over the best previous result [23], and an increase of 8.9% over the best previous result not using IR data. FOANet even surpassed human level accuracy by 2.9%.

The accuracy of FOANet drops to 85.26% when sparse network fusion is replaced by average fusion, emphasizing the importance of sparse network fusion even in domains with only one hand and no significant background changes. However, the accuracy of 85.26% is still better than the previous SOA, reaffirming the importance of focus of attention channels.

Table 6 gives per channel accuracies on NVIDIA test data. Similar to ChaLearn, we can see that the focused RGB flow field channel performs the best, followed by the focused depth channel. The general trend of focus channels being better than global channel is also evident here.

6. Conclusion and Future Work

Gestures are an important form of communication, and gesture recognition is an important application area for computer vision. Using the ChaLearn IsoGD and NVIDIA datasets as benchmarks, this paper shows recognition accuracy is significantly improved if convolutional channels are used not just to process different modes of data, but to focus attention within the scene. In particular, much of the information in gestures is in the hands, and channels that focus on the hands raise recognition rates from 67.71% to 82.07% on the IsoGD dataset, and from 83.8% to 91.28% on the more task-specific NVIDIA dataset.

We anticipate further improvements on FOANet. The current architecture does not address temporal fusion in a sophisticated way. Most gesture recognition networks fuse information over time using RNNs (e.g. [23, 31, 34]). Despite being susceptible to overfitting on small training sets, empirical data suggests RNNs nonetheless improve performance, and we anticipate adding them into FOANet.

Acknowledgements

This work was partially funded by the U.S. Defense Advanced Research Projects Agency and the U.S. Army Research Office under contract #W911NF-15-1-0459.

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. 1

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *Computer Vision-ECCV 2004*, pages 25–36, 2004. 5

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 5

[4] H. Cheng, L. Yang, and Z. Liu. Survey on 3d hand gesture recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9):1659–1673, 2016. 1

[5] 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. 3, 5

[6] H. J. Escalante, V. Ponce-López, J. Wan, M. A. Riegler, B. Chen, A. Clapés, S. Escalera, I. Guyon, X. Baró, P. Halvorsen, et al. Chalearn joint contest on multimedia challenges beyond visual analysis: An overview. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 67–73. IEEE, 2016. 1

[7] S. Escalera, X. Baró, J. Gonzalez, M. Á. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon. Chalearn looking at people challenge 2014: Dataset and results. In *ECCV Workshops (1)*, pages 459–473, 2014. 1

[8] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, I. Guyon, V. Athitsos, H. Escalante, L. Sigal, A. Argyros, C. Sminchisescu, et al. Chalearn multi-modal gesture recognition 2013: grand challenge and workshop summary. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 365–368. ACM, 2013. 1

[9] I. Guyon, V. Athitsos, P. Jangyodsuk, and H. J. Escalante. The chalearn gesture dataset (cgd 2011). *Machine Vision and Applications*, 25(8):1929–1951, 2014. 1

[10] I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner. Results and analysis of the chalearn gesture challenge 2012. In *Advances in Depth Image Analysis and Applications*, pages 186–204. Springer, 2013. 1

[11] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. J. Escalante. Chalearn gesture challenge: Design and first results. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–6. IEEE, 2012. 1

[12] H. Hasan and S. Abdul-Kareem. Human-computer interaction using vision-based hand gesture recognition systems: a survey. *Neural computing & applications*, 25(2), 2014. 1

[13] 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. 3

[14] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016. 3

[15] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017. 3

[16] 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. 2

[17] A. Kendon. *Gesture: Visible Action as Utterance*. Cambridge University Press, 2004. 1

[18] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Z. Liu. Chalearn2017_isolated_gesture. https://github.com/ZhipengLiu6/Chalearn2017_isolated_gesture, 2017. 3, 5

[20] Z. Liu, X. Chai, Z. Liu, and X. Chen. Continuous gesture recognition with hand-oriented spatiotemporal feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3056–3064, 2017. 3, 5

[21] D. McNeill. *Gesture & Thought*. University of Chicago Press, 2005. 1

[22] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, X. Cao, Z. Liu, X. Chai, Z. Liu, et al. Multimodal gesture recognition based on the resc3d network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3047–3055, 2017. 1, 2, 6

[23] 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. 1, 2, 8

[24] 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. 5

[25] 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. 1

[26] 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. 2, 5

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. 3

[28] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2

[29] 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. 1, 2, 5, 6

[30] 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. 1, 2

[31] 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. 1, 2, 6, 8

[32] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016. 2

[33] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 5

[34] L. Zhang, G. Zhu, P. Shen, J. Song, S. A. Shah, and M. Bennamoun. Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3120–3128, 2017. 1, 2, 6, 8

[35] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. https://arxiv.org/abs/1705.01389. 3, 5