Analysis of Deep Fusion Strategies for Multi-modal Gesture Recognition

Alina Roitberg*† Tim Pollert*† Monica Haurilet† Manuel Martin‡ Rainer Stiefelhagen†

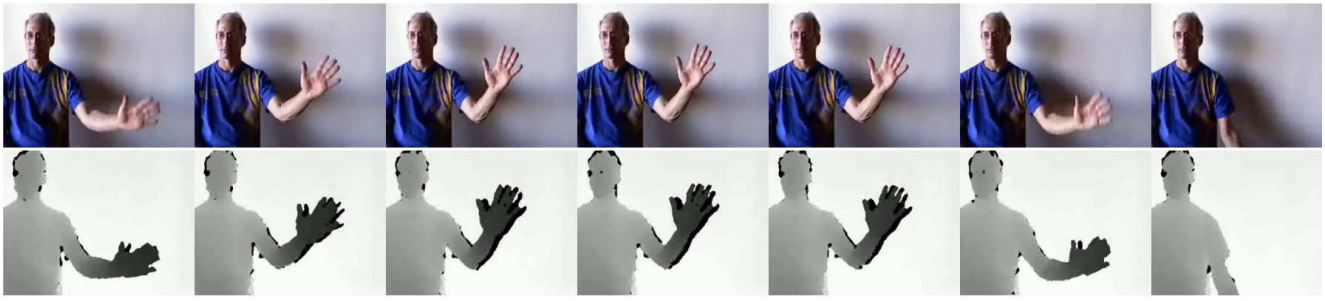

Figure 1: Example of a gesture in the IsoGD dataset, where a person is performing the sign for *five*. As we see, the data captured by an RGB camera (top) suffers from different illumination conditions *e.g.* the shadows produced by the light source to the left. However, the depth data (bottom) can have problems detecting the hand in case tit has the same depth as other objects close to it *e.g.* if the hand is almost touching the wall.

Abstract

Video-based gesture recognition has a wide spectrum of applications, ranging from sign language understanding to driver monitoring in autonomous cars. As different sensors suffer from their individual limitations, combining multiple sources has strong potential to improve the results. A number of deep architectures have been proposed to recognize gestures from e.g. *both color and depth data. However, these models conventionally comprise separate networks for each modality, which are then combined in the final layer (*e.g. *via simple score averaging). In this work, we take a closer look at different fusion strategies for gesture recognition especially focusing on the information exchange in the* intermediate *layers. We compare three fusion strategies on the widely used C3D architecture: 1) late fusion, combining the streams in the final layer; 2) information exchange in an intermediate layer using an additional convolution layer; and 3) linking information at multiple layers simultaneously using the* cross-stitch units*, originally designed for multi-task learning. Our proposed* C3D-Stitch *model achieves the best recognition rate, demonstrating the effectiveness of sharing information at earlier stages.*

*indicates equal contribution
†CV:HCI Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany
cvhci.anthropomatik.kit.edu
‡Fraunhofer IOSB, Karlsruhe, Germany iosb.fraunhofer.de

1. Introduction

Video-based gesture recognition provides an intuitive medium for human-machine interaction, attempting to detach computer input from conventional devices, such as mouse and keyboard (see example in Figure 1). Application areas of gesture recognition range from robotics [16] and understanding of sign language [3] to autonomous driving, where the driver can express his intention via gestures [15]. Multi-modality is an essential concept in such systems, since each sensor has its individual strengths and weaknesses [17]. For example, a large number of recognition models available for *color* images [10] are convenient for adaptation to other application domains (*e.g.* gestures) via transfer learning, although such RGB cameras are highly dependent on the illumination and fail at night. Depth sensors, on the other hand, are well-suited for realistic conditions for multiple reasons: they are less influenced by the light and mostly omit the surface texture (*e.g.* clothing), which is oftentimes irrelevant for gesture recognition and constitutes additional noise.

Deep neural networks achieve excellent results in many areas of computer vision and are also clear front-runners in the field of gesture recognition. Furthermore, successful methods in the current large-scale gesture recognition challenge "Chalearn Isolated Gesture Recognition" (IsoGD) are almost exclusively deep architectures adopted from the field of action recognition [22, 11, 12]. IsoGD is a large multi-modal dataset with videos of hand gestures, where each

sample covers both color and depth data. However, methods presented during the IsoGD challenge train separate neural networks for each data type and then use either a late fusion paradigm, e.g. averaging the prediction scores of the model, or limit the results to a single modality [22].

Despite a high correlation between the data streams, the possibilities of fusing the information at earlier stages has barely been explored in the area of gesture recognition. The main objective of our work, is to implement and systematically examine different strategies for sensor data fusion (*e.g*. color and depth information) for multi-modal gesture recognition with deep neural networks, covering both, the conventional late fusion and a variety of models based on earlier information exchange at *intermediate layers*.

Summary and Contributions Given the complementary nature of the input data, we argue, that gesture recognition models would benefit from fusion at intermediate layers. To validate our premise, we adopt the C3D architecture [20] based on 3D convolutions as our backbone model, which is widely used for gesture recognition [22, 11]. First, we train and evaluate separate single-modal networks and combine them afterwards with score averaging (*i.e*. late fusion) as our baselines (Figure 2). Next, we enhance the architecture with various building blocks for sharing the information at earlier stages of the network and evaluate their effect. We employ two different mechanisms at intermediate layers: 1) information exchange at a *single* intermediate layer and 2) fusion at *multiple* network layers simultaneously via *cross stitch units* [13]. In the first approach, we reduce the dimensionality of the two network outputs by half through an additional fusion layer with $1 \times 1 \times 1$ convolution filters. The output of this fusion layer is therefore a linear combination of the feature maps, which is further passed to a single *shared late network* (Figure 3). As our second strategy, we propose the *C3D-Stitch* architecture, leveraging the *cross stitch units*, which learn how to combine the activations of both networks with even less parameters, as a single weight is learned for each input feature map (Figure 4). Cross stitch units facilitate information exchange between the two sources, while keeping the original output dimensionality, and can therefore be *included at different depths of the network simultaneously*, so that the point of fusion is not be chosen by hand, as done in the first approach.

Our experiments on the ten most frequent gestures of the IsoGD dataset [22] demonstrate the effectiveness of exchanging information at intermediate layers in comparison to the single-modal baselines and the popular late fusion approach. The best recognition rate is achieved with the proposed *C3D-Stitch* network, where the fusion takes place at multiple layers at the same time.

2. Related Work

The field of gesture recognition is strongly influenced by progress in image analysis, as popular models for image classification are extended to be able to deal with image sequences by including a temporal dimension. Recent progress of deep learning methods has revolutionized the field, shifting the recognition paradigm from explicit definition of feature descriptors defined by hand [24, 25, 9] to *end-to-end learning* of good representations directly from visual input through Convolutional Neural Networks [10, 12, 22, 11], with a survey provided in [1].

Various modern gesture recognition architectures derive from methods of the related field of action recognition [2, 19, 14, 7]. Similarly to action recognition, in order to obtain a motion-based representation [22], optical flow is sometimes extracted from the image sequence and used instead of or in addition to the raw videos. There are different strategies for handling the temporal dimension, such as classifying image frames with conventional 2D CNNs and then averaging the results of all frames [19] or placing a recurrent neural network, such as an LSTM [6], on top of the CNN [14]. Motivated by the idea of making use of space-time features, Tran *et al*. [20] introduced the C3D architecture, which employs convolution layers with 3D kernels, which were also adapted in multiple other architectures [7, 2, 21].

Due to the growing interest in gesture recognition, various large-scale benchmarks were introduced in recent years, such as the ChaLearn Gesture Dataset (CGD) [5], which served as a basis for the large-scale Isolated Gesture Dataset (IsoGD) dataset [22, 23]. In the related recent gesture recognition challenge [22], the majority of proposed methods on gesture recognition adopt the C3D architecture as their backbone model [22]. We therefore also employ the C3D model as the core architecture in our framework and enhance it with building-blocks for mid-level fusion.

Fusing multiple modalities for deep-learning based gesture recognition is done with late fusion by the vast majority of previous approaches. They train individual networks for each modality, which are then joined via score averaging [22], using Support Vector Machines (SVMs) [11], using Canonic Correlation Analysis [12] or by a employing a voting strategy [4]. Despite the high correlation of information in the early stages of the multi-modal streams, such as in case of RGB and depth data, the research of deep fusion at intermediate network layers has been scarce so far. In this work, we aim to create a model which enables information sharing between the data sources at earlier stages in the model, by enhancing the C3D network with multiple fusion building blocks such as $1 \times 1 \times 1$ convolutions or cross-stitching units [13], which were originally designed for multi-task learning and additionally used to fuse different data streams for head pose estimation [18].

3. Fusion Strategies for Multi-modal Gesture Recognition

In this paper, we investigate various methods for deep multi-modal fusion in the context of hand gesture recognition. That is, given multiple video inputs (*i.e*. depth and color data), our goal is to identify the performed hand gesture, while combining the information from different streams in a beneficial way. While in the past, separately trained networks for each modality were joined via *late fusion*, we specifically focus on learning a shared representation at *intermediate* layers, which has been overlooked in the previous work.

To this intent, we employ the C3D [20] backbone architecture based on 3D convolutions, which has achieved excellent results for multi-modal gesture recognition (Section 3.1) and analyze the conventional late fusion approach (Section 3.2). We further evaluate merging at intermediate levels in the network and propose a straightforward method for linking the streams earlier via $1 \times 1 \times 1$ convolutions, which we examine at different network stages (Section 3.3). Finally, we propose a new architecture *C3D-stitch*, which learns how to combine the activations of both networks at multiple layers simultaneously by utilizing the cross stitch units (Section 3.4).

3.1. Backbone Architecture and Preprocessing

The backbone architecture of our pipeline is a Convolutional Neural Network (CNN) that employs spatio-temporal 3D kernels to handle the temporal dimension. We adopt the C3D architecture, as it has been most prominent on previous work for multi-modal gesture recognition[1]. Conceptually, our pipeline uses one C3D network for each modality. Since the dataset consists of color- and depth data, we train two C3D networks and examine various ways to link their information at different stages with the proposed fusion strategies.

Backbone Architecture. C3D consists of 8 convolutional layers, 5 pooling layers followed by two fully-connected layers and softmax normalization. The amount of filters increases from the first to the last convolutional layer starting with 64 filters, followed by 128, two 256 and three 512 convolutional layers, respectively. Four out of the five max pooling layers with kernel size of $2 \times 2 \times 2$ use a stride of 2 for increasing the receptive field and decreasing the amount of information to consider. The first pooling layer is an exception. In order to keep more temporal information, it only has a kernel size of $1 \times 2 \times 2$, with 1 denoting the temporal dimensions.

[1]We use the PyTorch implementation with its pre-trained weights on the Sports 1-M dataset provided in https://github.com/DavideA/c3d-pytorch.

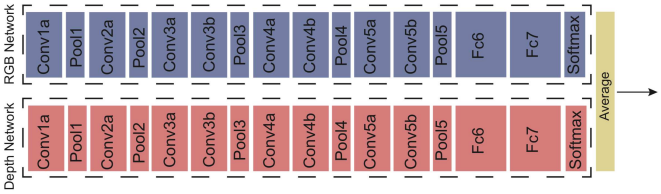

Figure 2: Overview of the late fusion model. This architecture consists of separate depth- and RGB-C3D-streams, where no interaction or information exchange is carried out between them. The fusion is carried out only in the final prediction layer (*i.e*. after the softmax normalization) where the confidences for each class is averaged between the two streams.

Spatial Alignment and Data Augmentation. As we aim to fuse the output of the *convolution* layers, correct spatial alignment between the feature maps of different modalities is important. However, the color- and depth frames of the IsoGD dataset are not perfectly aligned. In order to register the different views, we calculate the homography between the RGB and depth frames via multiple corresponding points. This operation aligns the views, therefore increasing their correlation. Following the original C3D implementation[20], we first rescale the videos to a resolution of 128×171 pixel. The input to the C3D network are then 16 cropped frames of 116×116 pixel. We employ random selection of the 16 frames and their cropping to achieve the desired resolution as our training data augmentation. At test-time, we compute center crops of the video frames.

Learning Setting. We train the model with a learning rate of 0.0001, momentum of 0.9 and a mini batch size of 10. We initialize the weights for both, color and depth streams, using a model pre-trained on the Sports-1M [8] dataset for large-scale action recognition.

3.2. Late Fusion Approach

Our first multi-modal strategy is *late fusion*, where we combine the outputs of the two networks though their last fully-connected layer by score averaging – a widely used method in gesture recognition. We investigate three different policies to train the model: 1) individual training of the two networks with two separate losses, 2) joint training of both networks in an end-to-end fashion, with a single loss estimated after averaging, and 3) a multi-step technique, where we first pre-train the networks on each modality individually and thereafter fine-tune them jointly. The learning parameters are identical to the backbone models that were trained separately for each modality (Section 3.1), except for the fine tuning phase of the network trained in multiple stages. An overview of the C3D network with the late fusion paradigm is illustrated in Figure 2.

Figure 3: Overview of the proposed intermediate fusion module via $1 \times 1 \times 1$ convolutions. We combine the two streams at different levels of the network *i.e.* at the second, third and fourth pooling layer. After the fusion module the two streams are merged to a single shared network using concatenation.

3.3. Mid-level Fusion with Shared Late Network

The main focus of this work are approaches, where the information exchange takes place at the feature maps level of the *intermediate* network layers, so that useful early feature correlations are taken into account. Our first intuition is to use separate streams at early layers and, then, fuse them into a joint model in a later stage (as depicted in Figure 3). A straight-forward fusion method is simply using $1 \times 1 \times 1$ convolutions followed by concatenation of the two output feature maps. The input shape for a single shared network of the next layer (after the fusion) should have the same shape as each of the two inputs to the fusion modules. Thus, we reduce the number of output filters by half in each $1 \times 1 \times 1$ convolution layer (*i.e.* we divide the number of filters by the number of streams). In other words, we employ the $1 \times 1 \times 1$ convolutions to decrease the dimensionality within the filter space. The final architecture therefore consists of three components: two early-stage networks corresponding to each individual modality and a shared network for the final stage, which leverages the shared input representation.

An important question when employing such a fusion scheme is selecting the point of fusion in the network, as we can select any convolution layer in the C3D architecture. Thus, we implement and compare different variants of the model, with fusion at different layers in the model. Figure 3 shows three model variants with the $1 \times 1 \times 1$ convolution layer before *conv_3a*, *conv_4a* and *conv_5a* of the shared network. We follow the same learning procedure as for the late fusion (Section 3.2). Furthermore, similar to Section 3.2, we evaluate both variants, with and without pre-training on the individual modalities.

3.4. Fusion on multiple Levels via Cross-stitch Units

Until now, we needed to manually select a certain stage in the model, at which the streams would be joined. In this section, we aim at building a model, which does not restrict, where the individual or joint learning takes place, and facilitates information exchange on *multiple layers at the same time*. We present a novel multi-stream model, which consists of individual C3D networks for each modality, which pass information to each other at each pooling and fully connected layer. In this architecture, the output of each of these layers is combined via a learned weighted average called cross-stitch units [13] (see overview of the *C3D-Stitch* model in Figure 4). In other words, at every stage all networks contribute to each other pairwise, while the extend of this contribution of foreign modalities is learned end-to-end.

We adapt the cross-stitch units building block, first used for multi-task learning, and utilize it for multi-modal fusion of single-task C3D networks. The cross-stitch units take two activation maps from both streams and pass a generated linear combination with learned weights to the next layer of each stream, respectively. In this way, the unit pieces together two new activation maps and passes them onto the next layer of the corresponding network.

More formally, let x_A, x_B be the feature maps of the two networks after layer ℓ (*e.g.* output of one of the pooling layers). The objective is to learn the linear combination $\hat{x}_A$, $\hat{x}_B$ of the two feature maps x_A, x_B:

$$\begin{bmatrix} \hat{x}_A^{i,j} \\ \hat{x}_B^{i,j} \end{bmatrix} = \begin{bmatrix} \alpha_{AA}^{\ell} & \alpha_{AB}^{\ell} \\ \alpha_{BA}^{\ell} & \alpha_{BB}^{\ell} \end{bmatrix} \begin{bmatrix} x_A^{i,j} \\ x_B^{i,j} \end{bmatrix}, \tag{1}$$

where i, j are location coordinates in the feature maps, while the α learned weights show the amount of information flow of each filter between the streams. The parameters α_{AA}, α_{BB} weight the information flow in the same modality, while α_{AB}, α_{BA} control the impact of the external modality stream on the current one. In other words, the α-values denote the degree of contribution of each pair of streams. A close to zero α_{AB} or α_{BA} value indicates that the amount of information shared between the modalities is low, while, high positive or low negative α_{AB} or α_{BA} weights are linked to a high amount of information exchange between the networks.

The core structure for each C3D model remains almost unchanged, as we extend its connections to the external network via cross-stitch units after each pooling layer and in-

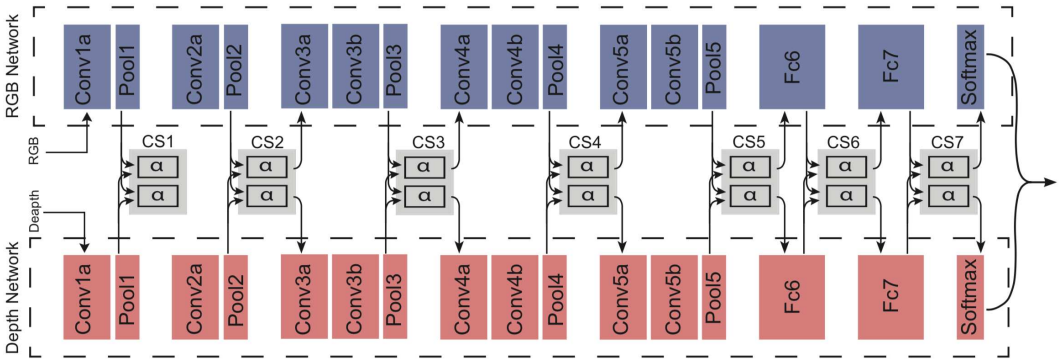

Figure 4: Overview of the proposed multi-layer fusion *C3D-Stitch* architecture. The model consists of two C3D streams, which pass each other information after each pooling and fully connected layer via cross-stitch units.

between the fully-connected layers. As the *C3D-Stitch* consists of two individual networks which actively share the information along the layers, the direct forward pass outputs two predictions. We therefore average the resulting softmax scores of both network and unify the prediction score. We follow the same learning procedure as for the late fusion (Section 3.2) and choose a cross-stitch layer learning rate of 0.01, similar to [13].

4. Experiments

We evaluate both our fusion policies and the single-stream baseline methods on the publicly available Isolated Gesture Dataset (IsoGD) [22, 23] for multi-modal gesture recognition. This benchmark consists of both color- and depth videos of 249 hand signs, where each video corresponds to a single isolated gesture. IsoGD is a large-scale dataset that provides a high variety of different gesture types of multiple applications ranging from sign language to diving and more specialized ones like gestures used for communication by Italians.

In this work, we focus on the potential of multi-layer fusion and conduct a systematic evaluation of various methods at different stages in the network. To this intent, we do not aim at improving the performance of current approaches, but selected a popular neural network often used in this task without any extensions such as skeleton extraction or hand cropping, which are often employed to improve the recognition rate.

In order to systematically evaluate fusion at different levels, we conduct our experiments on ten gestures, which are most frequent in the IsoGD dataset for mainly two reasons. First, the IsoGD dataset is highly unbalanced and considers classes, which occur only a few times in the dataset. This unbalance might influence the outcome of our evaluation, as the task gradually becomes few-shot learning. Secondly, due to the high computational cost of training on the entire dataset, we opt to include more experiments on a subset of the data instead of providing only a scarce analysis on the complete IsoGD. Thus, we evaluate our idea on the ten most frequent gestures from IsoGD, resulting in a dataset of 3711 gesture videos. We adopt the training, validation and test splits provided by the IsoGD benchmark.

4.1. Evaluation Metric

Following the evaluation procedure of the *Isolated Gesture Recognition Challenge* [22], we also use the recognition rate r as our default metric for comparing our fusion methods:

$$r = \frac{1}{n}\sum_{i=1}^{n}\delta(p(i), t(i)), \delta(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where $t(i)$ is the target of the i-th sample point, $p(i)$ is the prediction and n is the number of samples in our test set.

4.2. Late Fusion

Modality	Train. Proc.	Validation	Test
Baselines			
RGB	–	52.3	58.0
Depth	–	49.0	71.6
Late Fusion Methods			
RGB+Depth	separate	49.3	70.3
	combined	54.9	66.7
	sep.+comb.	**64.6**	**75.2**

Table 1: Results of C3D using late fusion compared with depth- and RGB-only. In this experiment, we evaluate different methods for late fusion where we: 1) train the models separately and combine the prediction only during testing; 2) train the depth and RGB-model together by averaging the cross entropy loss of both networks; 3) first train the networks separately and, then, fine-tune them together.

As a baseline, we first compare the commonly used late fusion approach with the single-stream models. We evaluate networks trained with three different training schemes described in Section 3.2: training two models separately, jointly, and a combination of both (first, they are trained separately and then, they are fine-tuned by averaging the losses). Table 1 illustrates the results of the experiment, clearly showing the benefit of multi-modal fusion. Training both networks jointly after single-modality pre-training leads to the best recognition rate of 75.2%, outperforming the depth-only model by over 3% and the RGB-only model by more than 17%.

4.3. Early and Mid-Fusion via $1 \times 1 \times 1$ convolutions

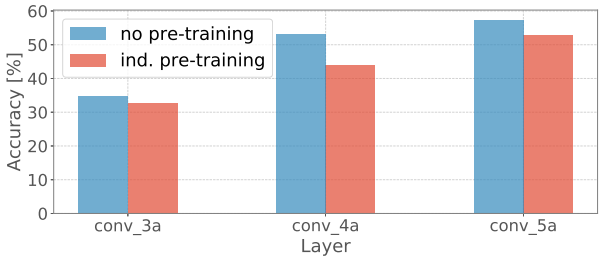

Figure 5: Validation accuracy of the fusion strategy using $1 \times 1 \times 1$ convolutions. We compare the performance between different placements of the fusion module. Furthermore, we differentiate between models that were first pre-trained individually and ones that were directly trained together.

Next, we explore the effect of the proposed mid-level fusion via $1 \times 1 \times 1$ convolutions (as described in Section 3.3). We add the fusion layer at different depths of the networks and report results for fusion at layers *conv_3a*, *conv_4a* and *conv_5a* of the C3D model illustrated in Figure 5. The position of the fusion layer has a great impact on the overall performance on the test set, ranging from 53.4% at the earliest layer to 78.6% at *conv_5a* layer (Table 2). We observe a clear trend for better classification results deeper in the network for both validation and test set. Still, information exchange via $1 \times 1 \times 1$ convolution at later stages surpasses the conventional late fusion method (*i.e.* at the softmax-layer) by over 3%.

4.4. Fusion via Cross-Stitching Units

Finally, we evaluate the effectiveness of the proposed *C3D-Stitch* model, where the networks share the information on multiple layers simultaneously. In Table 2, we perform extensive comparison between the *C3D-Stitch* network, late- and single-layer mid-level fusion approaches and the baseline methods. Similarly to previously considered methods, *C3D-Stitch* benefits from combining both, individual modality-specific pre-training and final joint optimization. As expected, our model outperforms single-model baselines by a large margin (17% for validation, 8.2% for testing) and are also more effective than the conventional late fusion strategy (1.4% for validation, 4.6% for test). Overall, the proposed *C3D-Stitch* network yields the best recognition rate of 79.8% . This outcome shows that modern multi-modal gesture recognition models can benefit more from sharing information between single convolution layer and late fusion. It further shows that it is helpful to employ a method like cross-stitch units that allow the network to learn end-to-end where and how much the different streams should interact with each other.

Modality	Ind. pre-train.	Layer	Validation	Test
Baselines				
RGB	–	–	52.3	58.0
Depth	–	–	49.0	71.6
Late Fusion Methods				
RGB+Depth	✗	softmax	54.9	66.7
	✓		64.6	75.2
$1 \times 1 \times 1$ Convolutions				
RGB+Depth	✗	conv_3a	32.8	42.7
	✓		34.7	53.4
	✗	conv_4a	44.1	64.8
	✓		53.2	70.5
	✗	conv_5a	52.8	75.2
	✓		57.4	78.6
Cross-stitch Units				
RGB+Depth	✗	multi-layer	56.6	77.1
	✓		**66.0**	**79.8**

Table 2: Results of C3D using the different fusion methods. We group our fusion methods into three categories: 1) late fusion where we combine the prediction of the networks after the final fully connected layer by simply averaging the confidences for each class; 2) early- and mid-level fusion using $1 \times 1 \times 1$ convolution layers to bridge the information between our two networks; 3) we apply cross-stitch units after each pooling and fully connected layer of the two C3D streams.

4.5. Learned Shared C3D-Stitch Representations

Networks with cross stitch units share the information through a linear combination of activation maps, where the corresponding weights are learned during training in an end-to-end fashion. In this section, we investigate the amount of information shared by the network as we take a look at the learned cross stitch units weights. The parameters α_C and α_D (Section 3.4) denote the weight each of the streams contribute to the output (C denotes color- and D depth network input). The weights are initialized in such a

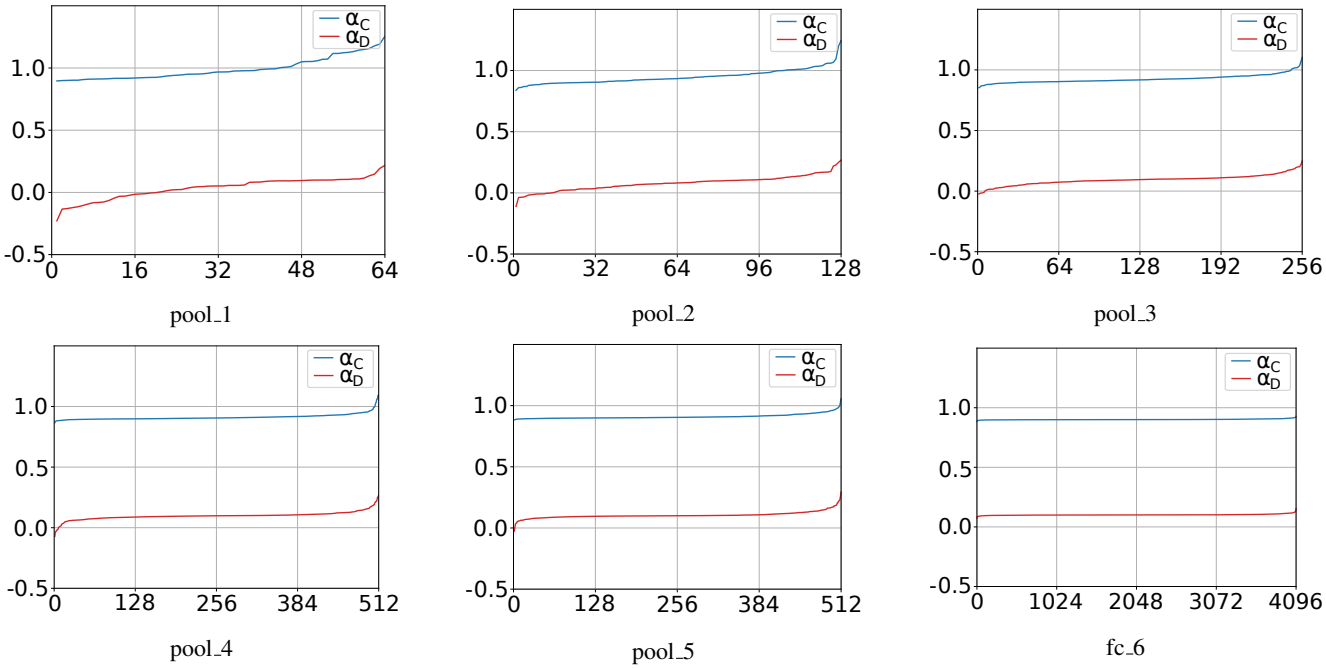

Figure 6: The sorted weights of the cross stitch units for different layers in our network, where we call α_C the weights of the RGB-C3D model, while α_D shows the importance of the depth architecture. Thus, the higher the values for α_C the more the network chooses the current features of the RGB stream, while higher α_D show a stronger preference in feature maps from the depth stream.

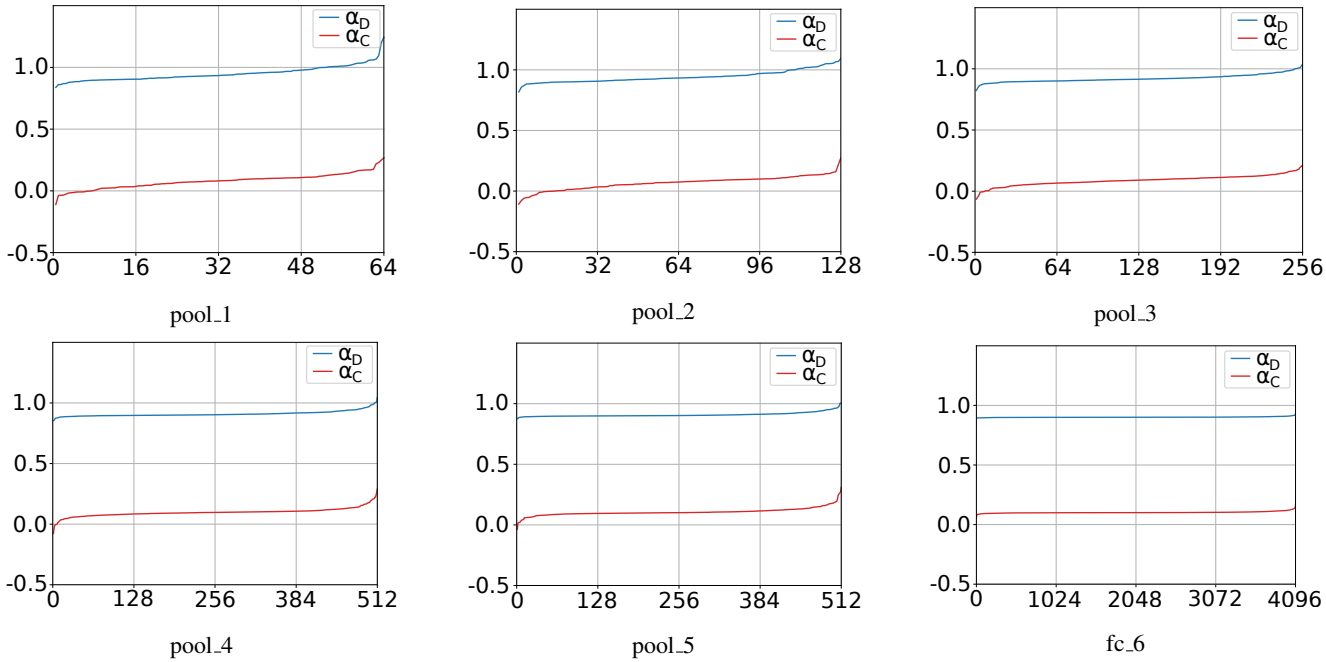

Figure 7: The sorted weights of the cross stitch units for different layers in our network, where we call α_C the weights of the RGB-C3D model, while α_D shows the importance of the depth architecture. Thus, the higher the values for α_C the more the network chooses information extracted from the RGB model, while higher α_D show a stronger preference in features of the current stream.

way that a small amount of information is shared between the two networks, as done in[13]. During training, the α values are learned to assure the optimal information sharing for the task.

We visualize the learned weights of the cross-stitch units in Figure 6 for the input to the color stream and in Figure 7 for the depth network. Both figures illustrate the sorted weights of each individual layer, where the cross stitch units are applied. We see in Figure 6, that while overall, internal features (in this case, color data), have a stronger contribution to the input of the next layer, we observe a clear mixture of the two modalities. The weights of the foreign depth network contains values of up to 0.25, while some α values of the color network have a value of over 1.0.

Overall, individual features of the same modality are weighted differently, *i.e*. our model has learned to select and share the most useful information. This exchange pattern is present along all layers, except for the last convolution layer *fc_6*, where the representation is still mixed, but the features seem to be weighted uniformly (around 0.9 for color and 0.1 for the foreign depth stream). We observe similar behavior for the depth sub-network (Figure 7), with active information exchange at all levels. In conclusion, these results demonstrate, that both the RGB and the depth model benefit from the knowledge sharing at multiple stages.

5. Conclusion

In this paper, we took a closer look at various CNN-based fusion strategies for multi-modal gesture recognition from videos. Going beyond the conventional late fusion paradigm, we specifically focus on merging the data at *intermediate* network layers. To achieve this, we proposed multiple enhancements for the popular C3D architecture: fusion in the middle of the network with an additional convolution layer and the *C3D-stitch* model, where the exchange happens at multiple layers simultaneously through the cross stitch units.

Our thorough analysis of different models for gesture recognition from color- and depth videos, has given three main findings: 1) we confirm our assumption, that gestures recognition benefits from multi-modality, as even simple multi-modal approaches surpass single-model ones; 2) we show, that involving mid-level features in the information exchange with an additional $1 \times 1 \times 1$ convolution layer further boosts the performance; 3) sharing the information at multiple layers simultaneously consistently outperforms single-layer fusion, which we demonstrate with our novel *C3D-stitch* architecture. The proposed *C3D-stitch* network achieves the best results with a performance increase of over 20% compared to the RGB-baseline model. Our experiments indicate, that multi-modal gesture recognition approaches could benefit further from utilizing earlier network layers for the information exchange between the streams.

Acknowledgements The research leading to this results has been partially funded by the German Federal Ministry of Education and Research (BMBF) within the PAKoS project.

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 *IEEE international conference on automatic face & gesture recognition (FG)*, pages 476–483. IEEE, 2017.

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[3] C. Dong, M. C. Leu, and Z. Yin. American sign language alphabet recognition using microsoft kinect. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 44–52, 2015.

[4] J. Duan, J. Wan, S. Zhou, X. Guo, and S. Z. Li. A unified framework for multi-modal isolated gesture recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(1s):21, 2018.

[5] I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner. Results and analysis of the chalearn gesture challenge 2012. In *International Workshop on Depth Image Analysis and Applications*, pages 186–204. Springer, 2012.

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

[7] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[8] 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.

[9] J. Konečnỳ and M. Hagara. One-shot-learning gesture recognition using hog-hof features. *The Journal of Machine Learning Research*, 15(1):2513–2532, 2014.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Y. Li, Q. Miao, K. Tian, Y. Fan, X. Xu, R. Li, and J. Song. Large-scale gesture recognition with a fusion of rgb-d data based on the c3d model. In *International Conference on Pattern Recognition (ICPR)*, pages 25–30. IEEE, 2016.

[12] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, and X. Cao. Multimodal gesture recognition based on the resc3d network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3047–3055, 2017.

[13] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

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

[15] E. Ohn-Bar and M. M. Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014.

[16] A. Roitberg, A. Perzylo, N. Somani, M. Giuliani, M. Rickert, and A. Knoll. Human activity recognition in the context of industrial human-robot interaction. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–10. IEEE, 2014.

[17] A. Roitberg, N. Somani, A. Perzylo, M. Rickert, and A. Knoll. Multimodal human activity recognition for industrial manufacturing processes in robotic workcells. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 259–266. ACM, 2015.

[18] A. Schwarz, M. Haurilet, M. Martinez, and R. Stiefelhagen. Driveahead-a large-scale driver head pose dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–10, 2017.

[19] 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.

[20] 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.

[21] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1510–1517, 2018.

[22] J. Wan, S. Escalera, G. Anbarjafari, H. Jair Escalante, X. Baró, I. Guyon, M. Madadi, J. Allik, J. Gorbova, C. Lin, et al. Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3189–3197, 2017.

[23] 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.

[24] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *IEEE Computer Society conference on computer vision and pattern recognition*, pages 379–385. IEEE, 1992.

[25] M. Yeasin and S. Chaudhuri. Visual understanding of dynamic hand gestures. *Pattern Recognition*, 33(11):1805–1817, 2000.