What have we learned from deep representations for action recognition?

Christoph Feichtenhofer[*]
TU Graz
feichtenhofer@tugraz.at

Axel Pinz
TU Graz
axel.pinz@tugraz.at

Richard P. Wildes
York University, Toronto
wildes@cse.yorku.ca

Andrew Zisserman
University of Oxford
az@robots.ox.ac.uk

Abstract

As the success of deep models has led to their deployment in all areas of computer vision, it is increasingly important to understand how these representations work and what they are capturing. In this paper, we shed light on deep spatiotemporal representations by visualizing what two-stream models have learned in order to recognize actions in video. We show that local detectors for appearance and motion objects arise to form distributed representations for recognizing human actions. Key observations include the following. First, cross-stream fusion enables the learning of true spatiotemporal features rather than simply separate appearance and motion features. Second, the networks can learn local representations that are highly class specific, but also generic representations that can serve a range of classes. Third, throughout the hierarchy of the network, features become more abstract and show increasing invariance to aspects of the data that are unimportant to desired distinctions (e.g. motion patterns across various speeds). Fourth, visualizations can be used not only to shed light on learned representations, but also to reveal idiosyncracies of training data and to explain failure cases of the system. ***This document is best viewed offline where figures play on click.***

1. Motivation

Principled understanding of how deep networks operate and achieve their strong performance significantly lags behind their realizations. Since these models are being deployed to all fields from medicine to transportation, this issue becomes of ever greater importance. Previous work has yielded great advances in effective architectures for recognizing actions in video, with especially significant strides towards higher accuracies made by deep spatiotemporal networks [2, 8, 32, 39, 40]. However, what these models actually learn remains unclear, since their compositional structure makes it difficult to reason explicitly about their learned representations. In this paper we use spatiotemporally regularized activation maximization [23, 25, 31, 38, 42] to visualize deep two-stream representations [32] and better understand what the underlying models have learned.

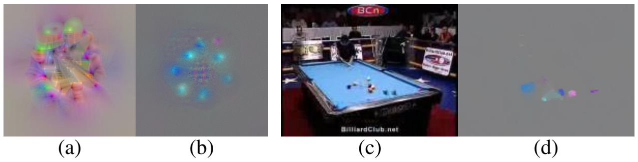

Figure 1. Studying a single filter at layer conv5_fusion: (a) and (b) show what maximizes the unit at the input: multiple coloured blobs in the appearance input (a) and moving circular objects at the motion input (b). (c) shows a sample clip from the test set, and (d) the corresponding optical flow (where the RGB channels correspond to the horizontal, vertical and magnitude flow components respectively). Note that (a) and (b) are optimized from white noise under regularized spatiotemporal variation. **Best viewed in Adobe Reader where (b)-(d) should play as videos.**

As an example, in Fig. 1 we highlight a single interesting unit at the last convolutional layer of the VGG-16 Two-Stream Fusion model [8], which fuses appearance and motion features. We visualize the appearance and motion inputs that highly activate this filter. When looking at the inputs, we observe that this filter is activated by differently coloured blobs in the appearance input and by linear motion of circular regions in the motion input. Thus, this unit could support recognition of the Billiards class in UCF101, and we show in Fig. 1c a sample Billiards clip from the test set of UCF101. Similar to emergence of object detectors for static images [1, 46], here we see the emergence of a spatiotemporal representation for an action. While [1, 46] automatically assigned concept labels to learned internal representations by reference to a large collection of labelled input samples, our work instead is concerned with visualizing the network's internal representations without appeal to any signal at the input and thereby avoids biasing the visualization via appeal to a particular set of samples.

Generally, we can understand deep networks from two viewpoints. First, the *architectural viewpoint* that considers a network as a computational structure (*e.g.* a directed acyclic graph) of mathematical operations in feature space (*e.g.* affine scaling and shifting, local convolution and pool-

[*] C. Feichtenhofer made the primary contribution to this work and therefore is listed first. Others contributed equally and are listed alphabetically.

ing, nonlinear activation functions, etc.). In previous work, architectures (such as Inception [36], VGG16 [33], ResNet [14]) have been designed by composing such computational structures with a principle in mind (e.g. a direct path for backpropagation in ResNet). We can thus reason about their expected predictions for given input and the quantitative performance for a given task justifies their design, but this does not explain how a network actually arrives at these results. The second way to understand deep networks is the *representational viewpoint* that is concerned with the learned representation embodied in the network parameters. Understanding these representations is inherently hard as recent networks consist of a large number of parameters with a vast space of possible functions they can model. The hierarchical nature in which these parameters are arranged makes the task of understanding complicated, especially for ever deeper representations. Due to their compositional structure it is difficult to explicitly reason about what these powerful models actually have learned.

In this paper we shed light on deep spatiotemporal networks by visualizing what excites the learned models using activation maximization by backpropagating on the input. We are the first to visualize the hierarchical features learned by a deep motion network. Our visual explanations are highly intuitive and provide qualitative support for the benefits of separating into two pathways when processing spatiotemporal information – a principle that has also been found in nature where numerous studies suggest a corresponding separation into ventral and dorsal pathways of the brain [9, 11, 24] as well as the existence of cross-pathway connections [19, 29].

2. Related work on visualization

The current approaches to visualization can be grouped into three types, and we review each of them in turn.

Visualization for given inputs have been used in several approaches to increase the understanding of deep networks. A straightforward approach is to record the network activities and sample over a large set of input images for finding the ones that maximize the unit of interest [1, 43, 46, 47]. Another strategy is to use backpropagation to highlight salient regions of the hidden units [22, 30, 31, 45].

Activation maximization (AM) has been used by backpropagating on, and applying gradient ascent to, the input to find an image that increases the activity of some neuron of interest [5]. The method was employed to visualize units of Deep Belief Networks [5, 15] and adopted for deep autoencoder visualizations in [21]. The AM idea was first applied to visualizing ConvNet representations trained on ImageNet [31]. That work also showed that the AM techniques generalize the deconvolutional network reconstruction procedure introduced earlier [43], which can be viewed as a special case of one iteration in the gradient based activation maximization. In an unconstrained setting, these methods can exploit the full dimensionality of the input space; therefore, plain gradient based optimization on the input can generate images that do not reflect natural signals. Regularization techniques can be used to compensate for this deficit. In the literature, the following regularizers have been applied to the inputs to make them perceptually more interpretable: $L2$ norms [31], total-variation norms [23], Gaussian blurring, and suppressing of low values and gradients [42], as well as spatial sifting (jittering) of the input during optimization, [25]. Backpropagation on the input has also been used to find salient regions for a given input [22, 35, 45], or to "fool" networks by applying a perturbation to the input that is hardly perceptible to humans [28, 38].

Generative Adversarial Networks (GANs) [12] provide even stronger natural image priors, for visualizing class level representations [26, 27] in the activation maximization framework. These methods optimize a high-dimensional code vector (typically fc_6 in AlexNet) that serves as an input to the generator which is trained with a perceptual loss [4] that compares the generater features to those from a pretrained comparator network (typically AlexNet trained on ImageNet). The approach induces strong regularization on the possible signals produced. In other words, GAN-based activation maximization does not start the optimization process from scratch, but from a generator model that has been trained for the same or a similar task [4]. More specifically, [26] trains the generator network on ImageNet and activation maximization in some target (ImageNet) network is achieved by optimizing a high-level code (*i.e.* fc_6) of this generator network. Activation maximization results produced by GANs offer visually impressive results, because the GAN enforces natural looking images and these methods do not have to use extra regularization terms to suppress extremely high input signals, high frequency patterns or translated copies of similar patterns that highly activate some neuron. However, the produced result of this maximization technique is in direct correspondence to the generator, the data used to train this model, and not a random sample from the network under inspection (which serves as a condition for the learned generative prior). Since we are interested in the raw input that excites our representations, we do not employ any generative priors in this paper. In contrast, our approach directly optimizes the spatiotemporal input of the models starting from randomly initialized noise image (appearance) and video (motion) inputs.

3. Approach

There are several techniques that perform activation maximization for image classification ConvNets [23, 25, 31, 38, 42] which have shown that features become more abstract when approaching deeper layers of image-based networks. We build on these methods for visualizing the hi-

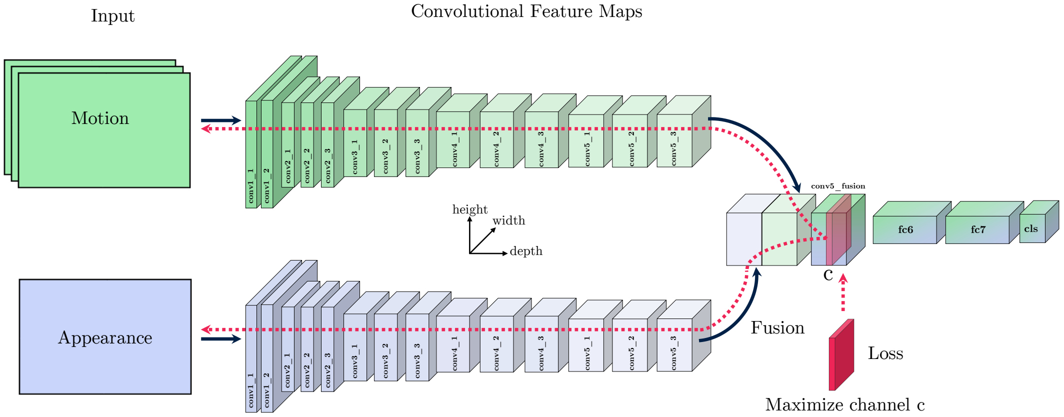

Figure 2. Schematic of our two-stream activation maximization approach (see Section 3 for details).

erarchical features learned by a deep motion network. In particular, we optimize in the spacetime domain to find the preferred spatiotemporal input of individual units in a Two-Stream Fusion model [8]. We formulate the problem as a (regularized) gradient-based optimization problem that searches in the input space. An overview of our approach is shown in Fig. 2. A randomly initialized input is presented to the optical flow and the appearance pathways of our model. We compute the feature maps up to a particular layer that we would like to visualize. A single target feature channel, c, is selected and activation maximization is performed to generate the preferred input in two steps. First, the derivatives on the input that affect c are calculated by backprogating the target loss, summed over all locations, to the input layer. Second, the propagated gradient is scaled by the learning rate and added to the current input. These operations are illustrated by the dotted red lines in Fig. 2. Gradient-based optimization performs these steps iteratively with an adaptively decreasing learning rate until the input converges. Importantly, during this optimization process the network weights are not altered, only the input receives changes. The detailed procedure is outlined in the remainder of this section.

3.1. Activation maximization

To make the above more concrete, activation maximization of unit c at layer l seeks an input $\mathbf{x}^* \in \mathbb{R}^{H \times W \times T \times C}$, with H being the height, W the width, T the duration, and C the color and optical flow channels of the input. We find $\mathbf{x}^*$ by optimizing the following objective

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}} \frac{1}{\rho_l^2 \hat{\mathbf{a}}_{l,c}} \langle \mathbf{a}_l(\mathbf{x}), e_c \rangle - \lambda_r \mathcal{R}_r(\mathbf{x}) \tag{1}$$

where $\mathbf{a}_l$ are the activations at layer l, e_c is the natural basis vector corresponding to the c^{th} feature channel, and $\mathcal{R}_r$ are regularization term(s) with weight(s) λ_r. To produce plausible inputs, the unit-specific normalization constant depends on ρ_l, which is the size of the receptive field at layer l (*i.e.* the input space), and $\hat{\mathbf{a}}_{l,c}$, which is the maximum activation of c recorded on a validation set.

Since the space of possible inputs that satisfy (1) is vast, and natural signals only occupy a small manifold of this high-dimensional space, we use regularization to constrain the input in terms of range and smoothness to better fit statistics of natural video signals. Specifically, we apply the following two regularizers, $\mathcal{R}_B$ and $\mathcal{R}_{TV}$, explicitly to the appearance and motion input of our networks.

3.2. Regularizing local energy

As first regularizer, $\mathcal{R}_B$, we enforce a local norm that penalizes large input values

$$\mathcal{R}_B(\mathbf{x}) = \begin{cases} N_B(\mathbf{x}) & \forall i,j,k : \sqrt{\sum_d \mathbf{x}(i,j,k,d)^2} \leq B \\ +\infty, & \text{otherwise.} \end{cases} \tag{2}$$

with $N_B(\mathbf{x}) = \sum_{i,j} \left(\sum_d \mathbf{x}(i,j,k,d)^2 \right)^{\frac{\alpha}{2}}$ and i,j,k are spatiotemporal indices of the input volume and d indexes either color channels for appearance input, or optical flow channels for motion input, B is the allowed range of the input, and α the exponent of the norm. Similar norms are also used in [23, 31, 42], with the motivation of preventing extreme input scales from dominating the visualization.

3.3. Regularizing local frequency

The second regularizer, $\mathcal{R}_{TV}$, penalizes high frequency content in the input, since natural signals tend to be dominated by low frequencies. We use a total variation regularizer based on spatiotemporal image gradients

$$\mathcal{R}_{TV}(\mathbf{x}; \kappa, \chi) = \sum_{ijkd} \left[\kappa \left((\nabla_x \mathbf{x})^2 + (\nabla_y \mathbf{x})^2 \right) + \chi (\nabla_t \mathbf{x})^2 \right], \tag{3}$$

where i,j,k are used to index the spatiotemporal dimensions of input $\mathbf{x}$, d indexes the color and optical flow channels of the input, and ∇_x, ∇_y, ∇_t are the derivative operators in the horizontal, vertical and temporal direction, respectively. κ is used for weighting the degree of spatiotemporal variation and χ is an explicit slowness parameter that accounts for the regularization strength on the temporal frequency. By varying $0 \leq \chi < \infty$ we can selectively penalize with respect to the slowness of the features at the input.

We now derive interesting special cases of (3) that we will investigate in our experiments:

- A purely spatial regularizer, $\kappa > 0; \chi = 0$ does not penalize variation over the temporal dimension, t. This choice produces reconstructions with unconstrained temporal frequency while only enforcing two-dimensional spatial smoothness in (3). This choice can be seen as an implicit low-pass filtering in the 2D spatial domain.
- An isotropic spatiotemporal regularizer, $\kappa = \chi; \kappa, \chi > 0$ equally penalizes variation in space and time. This can be seen as an implicit low-pass filtering in the 3D spatiotemporal domain.
- An anisotropic spatiotemporal regularizer, $\kappa \neq \chi; \kappa, \chi > 0$ allows balancing between space and time to *e.g.* visualize fast varying features in time that are smooth in space. The isotropic case above would bias the visualization to be smooth both in space and time, but not allow us to trade-off between the two.

Discussion. Purely spatial variation regularization is important to reconstruct natural images, examples of application include image/video restoration [44], feature inversion [23], or style transfer [17], or activation maximization [42] where a 2D Gaussian filter was applied after each maximization iteration to achieve a similar effect. Isotropic spatiotemporal regularization relates to multiple hand-designed features that operate by derivative filtering of video signals, examples include HOG3D [18], Cuboids [3], or SOEs [6]. Finally, anisotropic spatiotemporal regularization relates to explicitly modelling the variation in the temporal dimension. Larger weights χ in (3) stronger penalize the temporal derivative of the signal and consequently enforce low-pass characteristic such that it varies slowly in time. This is a well studied principle in the literature. For learning general representations from video in an unsupervised manner, minimizing the variation across time is seen both in biological, *e.g.* [10, 41], and artificial, *e.g.*, [13] systems. The motivation for such an approach comes from how the brain solves object recognition by building a stable, slowly varying feature space with respect to time [41] in order to model temporally contiguous objects for recognition.

In summary, the regularization of the objective, (1), combines (2) and (3): $\mathcal{R}_r(\mathbf{x}) = \mathcal{R}_B(\mathbf{x}) + \mathcal{R}_{TV}(\mathbf{x}; \kappa, \chi)$. Thus, $\mathcal{R}_r(\mathbf{x})$ serves to bias the visualizations to the space of natural images in terms of their magnitudes and spatiotemporal rates of change. Note that the three different special cases of the variational regularizer for the motion input allow us to reconstruct signals that are varying slowly in space, uniformly in spacetime and non-uniformly in spacetime.

3.4. Implementation details

For optimizing the overall objective, (1), we use ADAM that adaptively scales the gradient updates on the input by its inverse square root, while aggregating the gradients in a sliding window over previous iterations. We use the same initializations as in [23]. During optimization, we spatially sift (jitter) [25] the input randomly between 0 and the stride of the optimized layer. For all results shown in this paper, we chose the regularization/loss trade-off factors λ_r to provide similar weights for the different terms (2) - (3). We apply the regularizers separately to the optical flow and appearance input. The regularization terms for the appearance input are chosen to $\lambda_{B,\text{rgb}} = \frac{1}{HWB^\alpha}$ and $\lambda_{TV,\text{rgb}} = \frac{1}{HWV^2}$, with $V = B/6.5$, $B = 160$ and $\alpha = 3$, *i.e.* the default parameters in [23]. The motion input's regularization differs from that of appearance, as follows. In general, the optical flow is assumed to be smoother than appearance input; therefore, the total-variation regularization term of motion inputs has 10 times higher weight than the one for the appearance input. In order to visualize different speeds of motion signals, we use different weight terms for the variational regularizers of the motion input. In particular, to reconstruct different uniformly regularized spatiotemporal inputs we vary κ for penalizing the degree of spatiotemporal variation for reconstructing the motion input (we set $\chi = \kappa$ and only list the values for κ in the experiments). For anisotropic spatiotemporal reconstruction, we vary the temporal slowness parameter, χ and fix $\kappa = 1$. The values in all visualizations are scaled to min-max over the whole sequence for effectively visualizing the full range of motion.

4. Experiments

For sake of space, we focus all our experimental studies on a VGG-16 two-stream fusion model [8] that is illustrated in Fig. 2 and trained on UCF-101. Our visualization technique, however, is generally applicable to any spatiotemporal architecture. In the supplementary material[1], we visualize various other architectures: Spatiotemporal Residual Networks [7] using ResNet50 streams, Temporal Segment Networks [40] using BN-Inception [16] or Inception_v3 [37] streams, trained on multiple datasets: UCF101 [34], HMDB51[20] and Kinetics [2].

We plot the appearance stream input directly by showing an RGB image and the motion input by showing the optical flow as a video that plays on click; the RGB channels of this video consist of the horizontal, vertical and magnitude of the optical flow vectors, respectively. It is our impression that the presented flow visualization is perceptually easier to understand than standard alternatives (*e.g.* HSV encoding).

[1] http://feichtenhofer.github.io/action_vis.pdf

4.1. Emergence of spatiotemporal features

We first study the conv5_fusion layer (*i.e.* the last local layer; see Fig. 2 for the overall architecture), which takes in features from the appearance and motion streams and learns a local fusion representation for subsequent fully-connected layers with global receptive fields. Therefore, this layer is of particular interest as it is the first point in the network's forward pass where appearance and motion information come together. At conv5_fusion we see the emergence of both class specific and class agnostic units (*i.e.* general units that form a distributed representation for multiple classes). We illustrate both of these by example in the following.

Local representation of class specific units. In Fig. 1 we saw that some local filters might correspond to specific concepts that facilitate recognition of a single class (*e.g.* Billiards). We now reconsider that unit from Fig. 1 and visualize it under two further spatiotemporal regularization degrees, intermediate and fast temporal variation, in Fig. 3. (The visualization in Fig. 1 corresponds to slow motion.) Similar to Fig. 1, multiple coloured blobs show up in the appearance, Fig. 3a, and moving circular objects in the motion input (3b), but compared to Fig. 1, the motion is now varying faster in time. In Fig. 3d and 3c, we only regularize for spatial variation with unconstrained temporal variation, *i.e.* $\chi = 0$ in (3). We observe that this neuron is fundamentally different in the slow and the fast motion case: It looks for linearly moving circular objects in the slow spatiotemporal variation case, while it looks for an exploding, accelerating motion pattern into various directions in the temporally unconstrained (fast) motion case. It appears that this unit is able to detect a particular spatial pattern of motion, while allowing for a range of speeds and accelerations. Such an abstraction presumably has value in recognizing an action class with a degree of invariance to exact manner in which it unfolds across time. Another interesting fact is that switching the regularizer for the motion input, also has an impact on the appearance input (Fig. 3a *vs.* 3c) even though the regularization for appearance is held constant. This fact empirically verifies that the fusion unit also expects specific appearance when confronted with particular motion signals.

We now consider unit f004 at conv5_fusion in Fig. 4. It seems to capture some drum-like structure in the center of the receptive field, with skin-colored structures in the upper region. This unit could relate to the PlayingTabla class. In Fig. 4 we show the unit under different spacetime regularizers and also show sample frames from three PlayingTabla videos from the test set. Interestingly, when stronger regularization is placed on both spatial and temporal change (*e.g.* $\kappa = 10$, top row) we see that a skin colour blob is highlighted in the appearance and a horizontal motion blob is highlighted in the motion in the same area, which combined could capture the characteristic head motions of a drummer. In contrast, with less constraint on motion variation (*e.g.* $\chi = 0$, bottom row) we see that the appearance more strongly highlights the drum region, including hand and arm-like structures near and over the drum, while the motion is capturing high frequency oscillation where the hands would strike the drums. Significantly, we see that this single unit fundamentally links appearance and motion: We have the emergence of true spatiotemporal features.

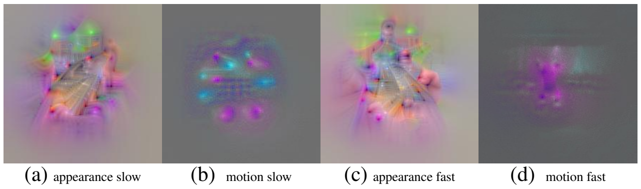

Figure 3. Studying the Billiards unit at layer conv5_fusion from Fig. 1. We now show what highly activates the filter in the appearance and in the motion input space using intermediate spatiotemporal variation regularization (a) and (b). Figs. (c) and (d) show what excites the filter when there is no restriction on the temporal variation of the input: The appearance, (3c) now also shows a black dot with skin-coloured surroundings at the top which might resemble a head and the motion filter (d) now detects exploding motion patterns (*e.g.* when the white ball hits the others after it has been accelerated by the billiard cue). **All videos play on click.**

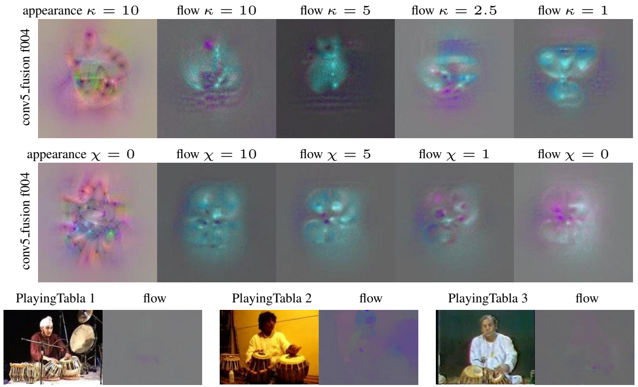

Figure 4. Specific unit at conv5_fusion. Comparison between isotropic and anisotropic spatiotemporal regularization for a single filter at the last convolutional layer. The columns show the appearance and the motion input generated by maximizing the unit, under different degrees of isotropic spatiotemporal (κ) and anisotropic spatiotemporal TV regularization (χ). The last row shows sample videos of appearance and optical flow from the PlayingTabla class.

Distributed representation of general units. In contrast to units that seem very class specific, we also find units that seem well suited for cross-class representation. To begin, we consider filters f006 and f009 at the conv5_fusion layer that fuses from the motion into the appearance stream, as shown in Fig. 5. These units seem to capture general spatiotemporal patterns for recognizing classes such as YoYo and Nunchucks, as seen when comparing the unit visualizations to the sample videos from the test set. Next, in Fig. 6, we similarly show general feature examples for the conv5_fusion layer that seem to capture general spatiotemporal patterns for recognizing classes corresponding to mul-

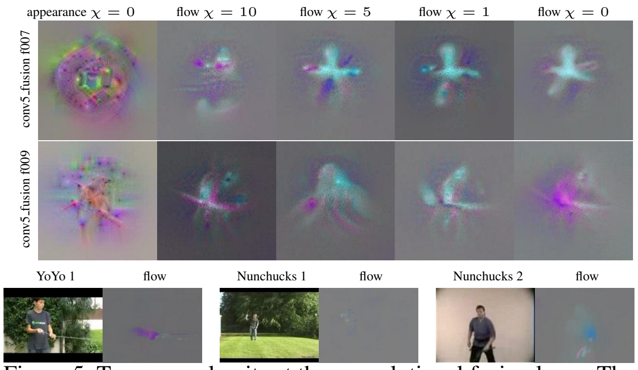

Figure 5. Two general units at the convolutional fusion layer. The columns show the appearance and the motion input generated by maximizing the unit, under different degrees of anisotropic spatiotemporal regularization (χ). The last row shows videos of 15 sample frames from the YoYo and Nunchucks classes.

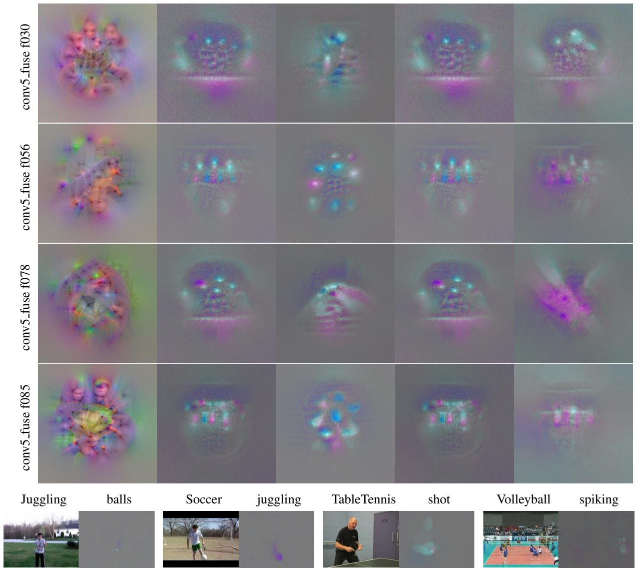

Figure 6. General units at the convolutional fusion layer that could be useful for representing ball sports. The columns show the appearance and the motion input generated by maximizing the unit, under different degrees of temporal regularization (χ). The last row shows sample videos from UCF101.

tiple ball sport actions such as Soccer or TableTennis. These visualizations reveal that at the last convolutional layer the network builds a local representation that can be both distributed over multiple classes and quite specifically tuned to a particular class (*e.g*. Fig. 4 above).

4.2. Progressive feature abstraction with depth

Visualization of early layers. We now explore the layers of a VGG-16 Two-Stream architecture [8]. In Fig. 7 we show what excites the convolutional filters of this architecture at the early layers of the network hierarchy. We use the anisotropic regularization in space and time that penalizes variation at a constant rate across space and varies according to the temporal regularization strength, χ, over time.

We see that the spatial patterns are preserved throughout various temporal regularization factors χ, at all layers. From the temporal perspective, we see that, as expected, for decreasing χ the temporal variation increases; interestingly, however, the directions of the motion patterns are preserved while the optimal motion magnitude varies with χ. For example, consider the last shown unit f36 of layer conv4_3 (bottom right filter in the penultimate row of Fig. 7). This filter is matched to motion blobs moving in an upward direction. In the temporally regularized case, $\chi > 0$, the motion is smaller compared to that seen in the temporally unconstrained case, $\chi = 0$. Notably, all these motion patterns strongly excite the same unit. These observations suggest that the network has learned speed invariance, *i.e*. the unit can respond to the same direction of motion with robustness to speed. Such an ability is significant for recognition of actions irrespective of the speed at which they are executed, *e.g*. being able to recognize "running" without a concern for how fast the runner moves. For a comparison of multiple early layer filters under isotropic spatiotemporal regularization please consider the supplementary material.

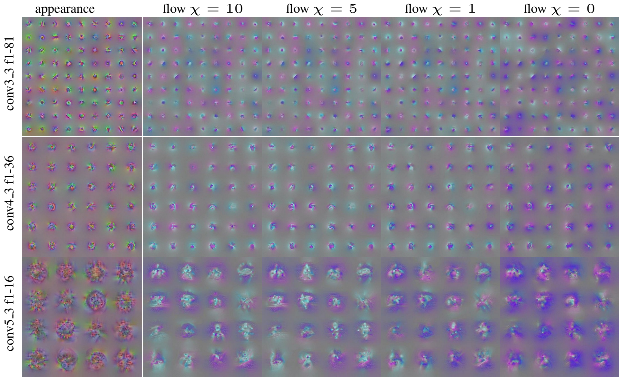

Figure 7. Two-stream conv filters under anisotropic regularization. We show appearance and the optical flow inputs for slowest $\chi = 10$, slow $\chi = 5$, fast $\chi = 1$, and fastest $\chi = 0$, temporal variation. Spatial regularization is constant.

Visualization of fusion layers. We now briefly re-examine the convolutional fusion layer (as in the previous Sect. 4.1). In Fig. 8, we show the filters at the conv5_fusion layer, which fuses from the motion into the appearance stream, while varying the temporal regularization and keeping the spatial regularization constant. This result is again achieved by varying the parameter χ in (3); visualizations of varying the regularization strengths isotropically (κ) are shown in the supplementary material. The visualizations reveal that these first 3 fusion filters at this last convolutional layer show reasonable combinations of appearance and motion information, a qualitative proof that the fusion model in [8] performs as desired. For example, the receptive field centre of conv5_fusion f002 seems matched to lip like appearance with a juxtaposed elongated horizontal structure, while the motion is matched to slight up and down motions of the elongation (*e.g*. flute playing). Once again, we also observe that the units are broadly tuned across temporal input variation (*i.e.* all the different inputs highly activate the same given unit).

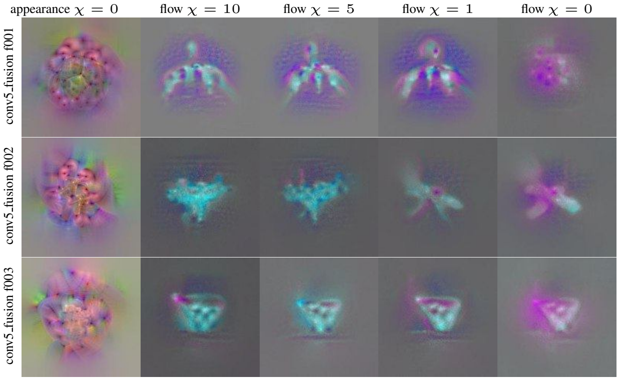

Figure 8. Visualization of 3 filters of the conv5_fusion layer. We show the appearance input and the optical flow inputs for slowest $\chi = 10$, slow $\chi = 5$, fast $\chi = 1$, and unconstrained $\chi = 0$, temporal variation regularization. The filter in row 2 could be related to the PlayingFlute class by locally filtering lips and a moving instrument (flute).

Visualization of global layers. We now visualize the layers that have non-local filters, *e.g.* fully-connected layers that operate on top of the convolutional fusion layer illustrated above. Fig. 9 and Fig. 10 show filters of the fully-connected layers 6 (fc_6) and 7 (fc_7) of the VGG-16 fusion architecture. In contrast to the local features above, we observe a holistic representation that consists of a mixture of the local units seen in the previous layer. For example, in Fig. 9 we see units that could combine features for prediction of VolleyballSpiking (top) PlayingFlute (centre) and Archery (bottom row); please compare to the respective prediction layer visualizations in the supplementary material. In Fig. 10 we see a unit that resembles the Clean and Jerk action (where a barbell weight is pushed over the head in a standing position) in the top row and another unit that could correspond to Benchpress action (which is performed in lying position on a bench). Notice how the difference in relative body position is captured in the visualizations, *e.g.* the relatively vertical vs. horizontal orientations of the regions captured beneath the weights, especially in the motion visualizations. Here, it is notable that these representations form something akin to a nonlinear (fc_6) and linear (fc_7) basis for the prediction layer; therefore, it is plausible that the filters resemble holistic classification patterns.

Finally, we visualize the ultimate class prediction layers of the architecture, where the unit outputs corresponds to different classes; thus, we know to what they should be matched. In Fig. 11, we show the fast motion activation of the classes Archery, BabyCrawling, PlayingFlute and CleanAndJerk (see the supplement for additional examples). The learned features for archery (*e.g.*, the elongated bow shape and positioning of the bow as well as the shooting motion of the arrow) are markedly distinct from those of the baby crawling (*e.g.*, capturing the facial parts of the baby appearance while focusing on the arm and head movement in the motion representation), and those of PlayingFlute (*e.g.* filtering eyes and arms (appearance) and moving arms below the flute (motion)), as well as those of CleanAndJerk and BenchPress (*e.g.* capturing barbells and human heads in the appearance with body motion for pressing ($\chi = 0$) and balancing ($\chi = 10$) the weight). Thus, we find that the class prediction units have learned representations that are well matched to their classes.

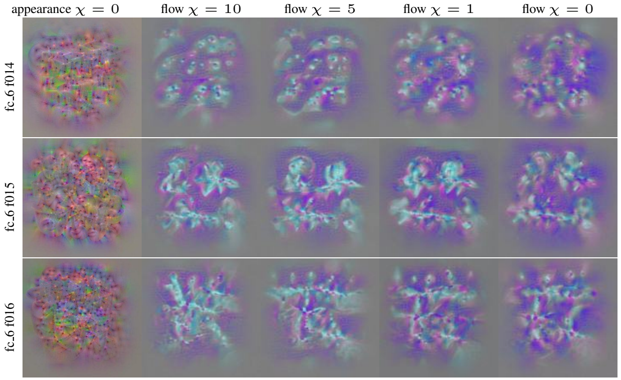

Figure 9. Visualization of 3 filters of the fc_6 layer under different temporal regularization. We show the appearance input and the optical flow inputs for slowest $\chi = 10$, slow $\chi = 5$, fast $\chi = 1$, and unconstrained (fastest) $\chi = 0$, temporal variation regularization. The filter shown in the first row could resemble the VolleyballSpiking class whereas the filter shown in the second row is visually similar to the unit for predicting PlayingFlute and the last row to the Archery class.

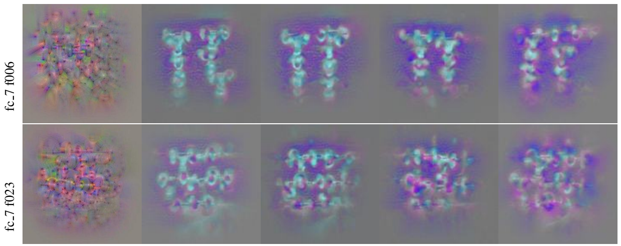

Figure 10. Visualization of 2 filters of the fc_7 layer under different temporal regularization strength χ. The filter shown in the first row is visually similar to the unit for predicting the Clean and Jerk class in the next layer, whereas the filter shown in the second row could resemble the BenchPress action.

4.3. Utilizing visualizations for understanding failure modes and dataset bias

Another use of our visualizations is to debug the model and reason about failure cases. In UCF101 15% of the PlayingCello videos get confused as PlayingViolin. In Fig. 12, we observe that the subtle differences between the classes are related to the alignment of the instruments. In fact, this is in concordance with the confused videos in which the Violins are not aligned in a vertical position.

In UCF101 the major confusions are between the classes

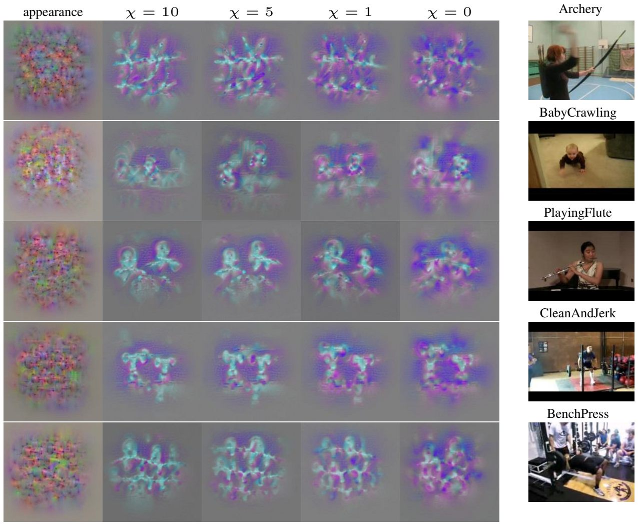

Figure 11. Classification units at the last layer of the network. The first column shows the appearance and the second to fifth columns the motion input generated by maximizing the prediction layer output for the respective classes, with different degrees of temporal variation regularization (χ). The last column shows 15 sample frames from the first video of that class in the test set.

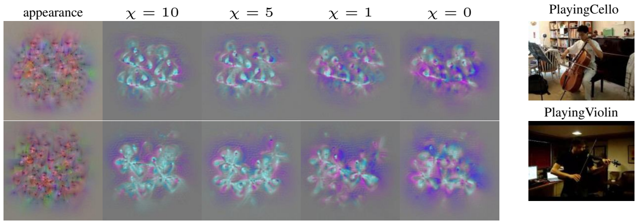

Figure 12. Explaining confusion for PlayingCello and PlayingViolin. We see that the learned representation focuses on the horizontal (Cello) and vertical (Violin) alignment of the instrument, which could explain confusions for videos where this is less distinct.

BrushingTeeth and ShavingBeard. In Fig. 13 we visualize the inputs that maximally activate these classes and find that they are quite similar, *e.g*. capturing a linear structure moving near the face, but not the minute details that distinguish them. This insight not only explains the confusion, but also can motivate remediation, *e.g*. focused training on the uncaptured critical differences (*i.e*. tooth brush *vs*. shaver).

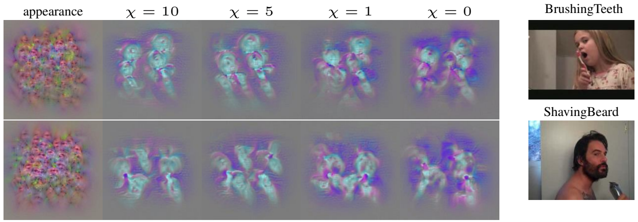

Figure 13. Explaining confusion for BrushingTeeth and ShavingBeard. The representation focuses on the local appearance of face and lips as well as the local motion of the tool.

Dataset bias and generalization to unseen data is important for practical applications. Two classes, ApplyEyeMakeup and ApplyLipstick are, even though being visually very similar, easily classified in the test set of UCF101 with classification rates above 90% (except for some obvious confusions with BrushingTeeth). This result makes us curious, so we inspect the visualizations in Fig. 14. The inputs are capturing facial features, such as eyes, and the motion of applicators. Interestingly, it seems that ApplyEyeMakeup and ApplyLipstick are being distinguished, at least in part, by the fact that eyes tend to move in the latter case, while they are held static in the former case. Here, we see a benefit of our visualizations beyond revealing what the network has learned – they also can reveal idiosyncrasies of the data on which the model has been trained.

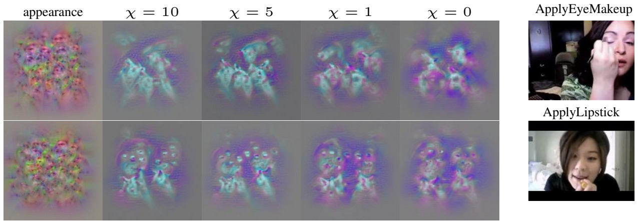

Figure 14. Classification units for ApplyEyeMakeup and ApplyLipstick. Surprisingly, the prediction neuron for ApplyLipstick gets excited by moving eyes at the motion input. Presumably because this resembles a peculiarity of the dataset which contains samples of the ApplyEyeMakeup class with eyes appearing static.

5. Conclusion

The compositional structure of deep networks makes it difficult to reason explicitly about what these powerful systems actually have learned. In this paper, we have shed light on the learned representations of deep spatiotemporal networks by visualizing what excites the models internally. We formulate our approach as a regularized gradient-based optimization problem that searches in the input space of a two-stream architecture by performing activation maximization. We are the first to visualize the hierarchical features learned by a deep motion network. Our visual explanations are highly intuitive and indicate the efficacy of processing appearance and motion in parallel pathways, as well as cross-stream fusion, for analysis of spatiotemporal information.

Acknowledgments

This work was partly supported by the Austrian Science Fund (FWF) under P27076, EPSRC Programme Grant Seebibyte EP/M013774/1, NSERC RGPIN 249910 and CFREF VISTA. Christoph Feichtenhofer is a recipient of a DOC Fellowship of the Austrian Academy of Sciences at the Institute of Electrical Measurement and Measurement Signal Processing, Graz University of Technology.

References

[1] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proc. CVPR*, 2017. 1, 2

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, 2017. 1, 4

[3] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *ICCV VS-PETS*, 2005. 4

[4] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016. 2

[5] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, University of Montreal, Jun 2009. 2

[6] C. Feichtenhofer, A. Pinz, and R. Wildes. Dynamically encoded actions based on spacetime saliency. In *Proc. CVPR*, 2015. 4

[7] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016. 4

[8] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proc. CVPR*, 2016. 1, 3, 4, 6

[9] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991. 2

[10] P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991. 4

[11] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992. 2

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[13] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun. Unsupervised feature learning from temporal data. In *Proc. ICCV*, 2015. 4

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 2

[15] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. 2

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015. 4

[17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, 2016. 4

[18] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *Proc. BMVC.*, 2008. 4

[19] Z. Kourtzi and N. Kanwisher. Activation in human mt/mst by static images with implied motion. *Journal of cognitive neuroscience*, 12(1):48–55, 2000. 2

[20] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proc. ICCV*, 2011. 4

[21] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In *Proc. ICML*, 2012. 2

[22] A. Mahendran and A. Vedaldi. Salient deconvolutional networks. In *Proc. ECCV*, 2016. 2

[23] A. Mahendran and A. Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *IJCV*, pages 1–23, 2016. 1, 2, 3, 4

[24] M. Mishkin, L. G. Ungerleider, and K. A. Macko. Object vision and spatial vision: two cortical pathways. *Trends in neurosciences*, 6:414–417, 1983. 2

[25] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. Google Research Blog. Retrieved June 20, 2015. https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html. 1, 2, 4

[26] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *NIPS*, 2016. 2

[27] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proc. CVPR*, 2017. 2

[28] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proc. CVPR*, 2015. 2

[29] K. Saleem, W. Suzuki, K. Tanaka, and T. Hashikawa. Connections between anterior inferotemporal cortex and superior temporal sulcus regions in the macaque monkey. *Journal of Neuroscience*, 20(13):5083–5101, 2000. 2

[30] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016. 2

[31] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2014. 1, 2, 3

[32] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 2

[34] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions calsses from videos in the wild. Technical Report CRCV-TR-12-01, 2012. 4

[35] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR Workshop*, 2015. 2

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 2

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015. 4

[38] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proc. ICLR*, 2014. 1, 2

[39] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri.

Learning spatiotemporal features with 3D convolutional networks. In *Proc. ICCV*, 2015. 1

[40] 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 *ECCV*, 2016. 1, 4

[41] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002. 4

[42] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *ICML Workshop*, 2015. 1, 2, 3, 4

[43] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. 2

[44] H. Zhang, J. Yang, Y. Zhang, and T. S. Huang. Non-local kernel regression for image and video restoration. In *Proc. ECCV*, 2010. 4

[45] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *Proc. ECCV*, 2016. 2

[46] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *Proc. ICLR*, 2014. 1, 2

[47] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proc. CVPR*, 2016. 2