Hallucinate the motion of the image based on priors learned from the videos

# Im2Flow: Motion Hallucination from Static Images for Action Recognition

幻觉

Ruohan Gao
UT Austin
rhgao@cs.utexas.edu

Bo Xiong
UT Austin
bxiong@cs.utexas.edu

Kristen Grauman
UT Austin
grauman@cs.utexas.edu

learn a prior

## Abstract

Appearance is deprived of the rich dynamic structure and motions

*Existing methods to recognize actions in static images take the images at their face value, learning the appearances—objects, scenes, and body poses—that distinguish each action class. However, such models are deprived of the rich dynamic structure and motions that also define human activity. We propose an approach that hallucinates the unobserved future motion implied by a single snapshot to help static-image action recognition. The key idea is to learn a prior over short-term dynamics from thousands of unlabeled videos, infer the anticipated optical flow on novel static images, and then train discriminative models that exploit both streams of information. Our main contributions are twofold. First, we devise an encoder-decoder convolutional neural network and a novel optical flow encoding that can translate a static image into an accurate flow map. Second, we show the power of hallucinated flow for recognition, successfully transferring the learned motion into a standard two-stream network for activity recognition. On seven datasets, we demonstrate the power of the approach. It not only achieves state-of-the-art accuracy for dense optical flow prediction, but also consistently enhances recognition of actions and dynamic scenes.*

## 1. Introduction

Video-based action recognition has long been an active research topic in computer vision [9, 35, 49, 78, 79], with many recent methods employing deep Convolutional Neural Networks (CNNs) [65, 40, 37, 86, 70, 12, 81]. Regardless of the approach, most methods rely on two crucial and complementary cues: appearance and motion. Motion is usually represented by (stacked) optical flow or flow-based descriptors [65, 12, 9, 52, 81, 19], localized spatio-temporal descriptors [48, 82] or trajectories [78, 79].

*Static-image action recognition* instead requires the system to identify the activity taking place in an individual photo [25]. The problem is of great practical interest for organizing photo collections (e.g., on the Web, social media, stock photos) based on human behavior and events. Yet, it presents the additional challenge of understanding activity in the absence of motion information.

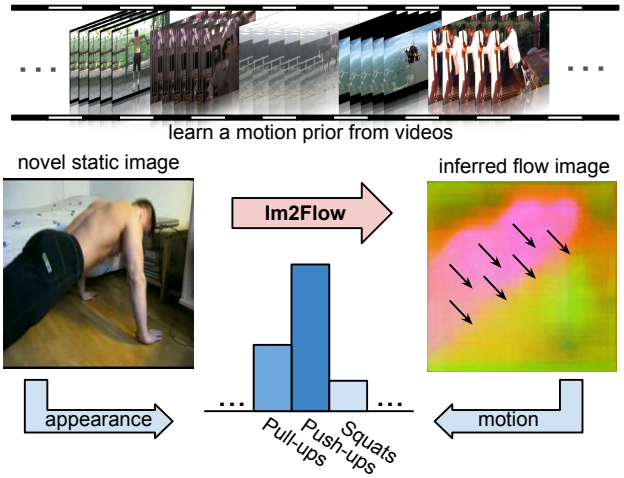



Figure 1. Our system first learns a motion prior by watching thousands of video clips containing various actions. Then, given a static image, the system translates the observed RGB image into a flow map encoding the inferred motion for the static image. Finally, we combine the appearance of the original static image and the motion from the inferred flow to perform action recognition.

Combine flow map and the rgb information to classify

Or is the motion really absent? A static snapshot can reveal the motions that are likely to occur next, at least for human viewers. Indeed, neuroscientists report that the medial temporal/medial superior temporal cortex—one of the main brain regions engaged in the perceptual analysis of visual motion—is also involved in representing implied motion from static images [44]. Over many years of observations, humans accumulate visual experience about how things move in the world. Given a single static image, not only can we interpret the instantaneous semantic content, but also we can anticipate what is going to happen next, e.g., based on human poses and object configurations present in the image. This suggests that a human viewer can leverage the *implied* motion to help perceive actions in static images. For example, given a static image as shown in Fig. 1, expecting that the person's back is going to either move up or down may aid the recognition of push-ups.

Learn prior from the video

We propose an approach for static-image action recognition that is inspired by this notion of visual dynamics accumulated from past temporal observations. The main idea is to acquire from videos a model for how objects and people move, then embed the resulting knowledge into a representation for individual images. In this way, even when limited to just one moment of observation (a single image), action recognition can be informed by the anticipated dynamics. In particular, we train a deep network to learn a motion prior from a large set of unlabeled videos, and then transfer the learned motion from videos to static images to hallucinate their motion. We devise an effective encoding for optical flow (Sec. 3.1) as well as an encoder-decoder network to learn the motion prior from videos (Sec. 3.2). Finally, we leverage the predicted motion to aid action recognition, by combining the appearance from the original static image and the motion from the inferred flow.

On seven challenging datasets for recognition of actions and dynamic scenes, our approach yields a significant accuracy boost by incorporating the hallucinated motion. Importantly, we also demonstrate that our approach is beneficial even in the case where the motion prior training videos do not contain the same actions as the static images.

The potential to extent to wild images

We make two main contributions. First, we formulate motion prediction as a novel image-to-image translation framework, and achieve state-of-the-art performance on dense optical flow prediction from static images, improving substantially on prior formulations for flow estimation [77, 58]. Secondly, we explore how implied motion aids action recognition. We show how injecting inferred motion into a standard two-stream network achieves significant gains compared to the one-stream counterpart, and we obtain state-of-the-art accuracy for multiple benchmarks.

## 2. Related Work

Our work relates to action recognition, visual anticipation, and image-to-image translation.

**Action Recognition** Video-based action recognition is a well-studied problem. Various video representations have been proposed to utilize both appearance and motion, including hand-crafted local spatio-temporal features [48, 82, 78, 79]; mid-level features [61, 32, 80]; and deeply-learned features [65, 40, 86, 12, 70]. Recent work aims to model long term temporal structure, via recurrent neural networks [86, 8], ranking functions [13], or pooling across space and time [19].

In static images, due to the absence of temporal information, various high-level cues are exploited, e.g., human body or body parts [69, 55, 84], objects [59, 84, 63], human-object interactions [7, 2], and scene context [26, 20]. See [25] for a comprehensive survey. Our work also targets action recognition in static images, but, unlike any of the above, we equip static images with dynamics learned from videos. To our knowledge, the only prior static-image approach to explicitly leverage video dynamics is [4]. However, whereas [4] leverages video to augment training images for the low-shot learning scenario, our method leverages video as a motion prior that enhances *test* observations. Our experiments compare the two methods.

**Visual Anticipation** Our work is also related to a wide body of work on visual future prediction [87, 43, 76, 58, 17, 77, 75, 72]. Most closely related are methods to predict optical flow from an image [76, 77, 58]. As we show in results, our formulation offers more accurate predictions. Furthermore, unlike any prior flow prediction work, we propose to integrate implied motion learned from thousands of unlabeled videos with action recognition. Note that while flow and depth are closely related problems, methods like DeepStereo [16] or DeepMorphing [36] assume two viewpoints as input to predict intermediate views. Other prediction tasks consider motion trajectories [75, 50] and human body poses [18, 3]. Another growing line of work aims to predict future frames in video [60, 68, 56, 83, 14, 73, 74, 71]. Their goal is to generate images of good visual quality to illustrate the "plausible futures", or potentially improve representation learning [68, 14].

In general, this line of work treats prediction as the end goal: e.g., predicting optical flow [77]; people's trajectories in a parking lot [43]; car movements on streets [76]; or subsequent high-level events [87, 72]. In contrast, our objective is to infer motion as an auxiliary cue for action recognition in static images. Our idea bridges static-image recognition with video-level action understanding by transferring a motion prior learned from videos to images.

**Image-to-Image Translation** Our technical solution for flow inference relates broadly to prior efforts to map an input pixel matrix directly to an output matrix. Early work in so-called "image-to-image translation" can be traced back to image analogies [28], where a nonparametric texture model is generated from a single input-output training image pair. Recent work uses generative adversarial models [23] to perform image-to-image translation, with impressive results [31, 54, 90]. Our dense optical flow prediction approach can be seen as a distinct image-to-image translation problem, where we encode the output motion space as a single "image" matrix.

## 3. Approach

Our goal is to learn a motion prior from videos, and then transfer the motion prior to novel static images to enhance recognition. We first discuss how we encode optical flow for more reliable prediction (Sec. 3.1). Then we present our Im2Flow network for motion prediction (Sec. 3.2). Finally, we describe how we use the predicted motion to help static-image action recognition (Sec. 3.3).

Two images, horizontal and vertical respectively, to represent the optical flow.

### 3.1. Motion Encoding

Optical flow encodes the pattern of apparent motion of objects in a visual scene, and it is the most direct and common motion information used for action recognition. (Stacked) optical flow is frequently used as input to deep methods [65, 12, 70, 81]. Often optical flow is represented by two separate grayscale images (matrices) that encode the quantized horizontal and vertical displacements. However, since state-of-the-art pre-trained deep networks [45, 66, 27] take a 3-channel RGB image as input, alternative encodings augment the two displacement maps with a third channel containing either the flow magnitude [8] or all zeros [86]. In such encodings, the third channel stores either redundant or useless information. Another approach is to encode the flow as an RGB image designed for visualization [1, 33], but this mapping is not reversible to obtain the motion vectors. Prior work quantizes flow vectors to 40 clusters in order to treat flow estimation as classification [77], but this has the drawback of providing only coarse flow estimates insufficient for recognition.

Our preliminary tests with the existing encodings confirmed these limitations (see Supp.), leading us to develop a new encoding scheme well-suited to motion prediction via regression. Directly predicting the optical flow $(u,v)$ vector for each pixel in a static image is a highly under-constrained problem, and we hypothesize that detangling flow direction and strength may present an easier learning task. Therefore, we decouple the optical flow into the motion angle $\theta \in [0, 2\pi]$ and magnitude $\mathcal{M}$. As we will describe in Sec. 3.2, we formulate flow prediction as a pixel-wise regression problem. Hence, direct prediction of $\theta$ is inappropriate because the angle in the coordinate system is circular (e.g., $2\pi$ is the same as 0). Therefore, we further divide $\theta$ into a horizontal direction and a vertical direction, represented by $\cos(\theta)$ and $\sin(\theta)$, respectively. We encode optical flow as a single 3-channel flow image $\mathcal{F}$:

Detangle flow direction and strength

Circular to non-circular

$$\mathcal{F}_1 = \sin(\theta) = \frac{v}{\mathcal{M}}; \quad \mathcal{F}_2 = \cos(\theta) = \frac{u}{\mathcal{M}}; \quad \mathcal{F}_3 = \mathcal{M}. \quad (1)$$

where $\mathcal{F}_i$ denotes the $i$-th channel.

Our motion encoding scheme has the following benefits: 1) It disentangles the convolved $(u,v)$ vector into three separate components, each indicating one important factor of motion, namely vertical direction, horizontal direction, and motion magnitude. This makes the high-dimensional motion prediction problem more factored; 2) It makes the regression of angle feasible, because $\sin(\theta)$ and $\cos(\theta)$ are non-circular and lie in the range of $[-1, 1]$; 3) Encoding motion as a 3-channel image makes its usage efficient, convenient, and suitable for our framework, defined next.

### 3.2. Im2Flow Network

Let $X$ be the domain of static images containing an action, and let $Y$ consist of the corresponding flow images encoded as defined above. Our goal is to learn a mapping $G : X \rightarrow Y$ that will infer flow from an individual image. During training, we are given "labeled" pairs $\{x_i, y_i\}_{i=1}^N$ consisting of video frames $x_i$ and the true flow maps $y_i$ computed from surrounding frame(s) in the source video. We use the optical flow algorithm of [53] to automatically generate $y$ for training data, since it offers a good balance between speed and accuracy.[1] To mitigate the effects of noisy flow estimates stemming from realistic training videos (i.e., we train with YouTube data, cf. Sec. 4), following [77], we average the optical flow of five future frames for each training image $x_i$ to compute the target $y_i$.

Training data

We devise a convolutional neural network (CNN) called Im2Flow to obtain $G$. Our Im2Flow network is an encoder-decoder, which takes a static image as input and outputs the predicted flow image $\hat{y} = G(x) = \mathcal{F}$. We adapt the U-Net architecture from [31] with some modifications, as illustrated in Fig. 2. Both the encoder and decoder use modules of the form Convolution-BatchNorm-ReLU [30]. We use dilated convolutions [85] in the encoder. Dilated convolutions exponentially increase their receptive field size and maintain spatial resolution, which can capture long-range spatial dependencies. The decoder is an up-convolutional network that generates the predicted flow image. Skip connections connect the encoder and decoder to directly shuffle low-level information across the network, which is important for our dense motion prediction problem. See Supp. for the details of the complete architecture.

Our Im2Flow network minimizes the combination of two losses: a pixel error loss and a motion content loss:

$$L = L_{pixel} + \lambda L_{content}^{\phi,j}. \quad (2)$$

The pixel loss measures the agreement with the true flow:

$$L_{pixel} = \mathbb{E}_{p,q \in \{x_i, y_i\}_{i=1}^N} [||y_i - G(x_i)||_2] \quad (3)$$

for all pixels $p, q$ in the training images. It requires the training flow vectors to be accurately recovered.

The motion content loss enforces that the predicted motion image preserve high level motion features. It follows the spirit of previous perceptual loss functions [39], but here for the sake of regularizing to realistic motion patterns. To represent realistic motion, we fine-tune an 18-layer ResNet [27] (pre-trained on ImageNet) for action classification on the UCF-101 dataset [67] using flow images as input. The motion content loss network $\phi$ is the resulting fine-tuned network. Then, we compute the $L_2$ loss on the activation maps extracted from the loss network $\phi$ for the predicted flow and ground-truth flow images. Hence, apart from encouraging the pixels of the output flow $G(x)$ to exactly match the pixels of the target flow $y$, we also encour-

[1] Accurate estimation of optical flow from real-world videos is a challenging problem on its own and is intensively studied in the literature [62, 15, 46, 29]. More accurate optical flow estimation could further improve the Im2Flow framework.

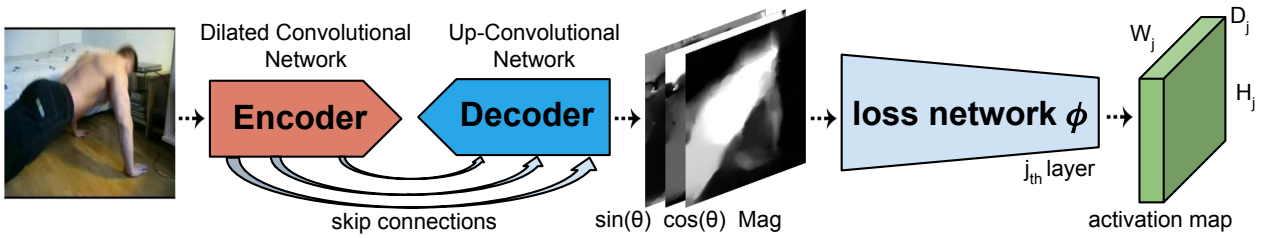



Figure 2. The network architecture of our Im2Flow framework. The network is an encoder-decoder that takes a static image as input and generates the corresponding 3-channel flow map $\mathcal{F}$ as output. Our training objective is a combination of the $L_2$ loss in the pixel space and in the deep feature space. A motion content loss network encourages the predicted flow image to preserve high-level motion features.

age them to have similar high-level motion representations as computed by the loss network.

Note that our approach can operate with both unlabeled and labeled video. The supervision from action labels used in the motion content loss network slightly helps to transfer motion from videos to images, and enhances the results on static-image action recognition. As we show in the ablation study in Supp., our approach maintains substantial gains even if we completely remove supervision in our framework, i.e., our approach can learn solely from unlabeled video. Moreover, even if we do learn from labeled video, the test data is not assured to be from the same actions (cf. Sec. 4.2).

Let $\phi_j(x)$ be the activations of the $j$-th layer of the network $\phi$ when processing the image $x$, and suppose $\phi_j(x)$ has a feature map of shape $D_j \times H_j \times W_j$. The motion content loss is the (squared, normalized) Euclidean distance between the feature representations:

$$L_{content}^{\phi,j} = \frac{1}{D_j \times H_j \times W_j} \mathbb{E}_{p,q \in \{x_i, y_i\}_{i=1}^{N}} \left[ ||\phi_j(y_i) - \phi_j(G(x_i))||_2 \right]. \quad (4)$$

We adjust optimization of the network to account for structure in our problem. The angles for pixels of very low motion strength are less crucial, because these pixels tend to correspond to static scenes in the image, e.g., the bed and wall in the input image in Fig. 2. Therefore, the motion directions of these pixels are not as meaningful, and they usually originate from the camera motion. To require the network to focus more on predicting directions of the pixels that actually move, we weight the gradients of the first two channels by their motion magnitude. The network uses the weighted gradients to perform back-propagation. The weighting process forces the network to emphasize predictions on moving pixels.

The above architecture was the most effective among other alternatives we explored. In particular, in preliminary experiments, we implemented a conditional generative adversary network (cGAN) to perform flow prediction, inspired by [31, 90]. In principle, such a GAN might handle multiple modes of motion ambiguity and predict flow images that encode realistic motion. However, we found that cGAN only helped to generate motion images of similar *visual style* to the ground-truth flow maps: the color patterns were similar to the ground-truth, but the encoded motion was less accurate. Because the cGAN discriminator cares about differentiating real and fake outputs, the approach seems better suited to problems requiring output images of good visual quality, as opposed to our task, which requires precise pixel-wise estimates.

### 3.3. Action Recognition with Implied Motion

Recall our goal is twofold: to produce accurate flow maps for static images, and to explore their utility as auxiliary input for static image action recognition. For the latter, we adopt the popular and effective two-stream CNN architecture [65] that is now widely used for CNN-based action recognition with videos [5, 8, 21, 86, 68, 12, 81, 19]. The two-stream approach is designed to mimic the pathways of the human visual cortex for object recognition and motion recognition [22]. Namely, the method decomposes video into spatial (RGB frames) and temporal (optical flow) components. These two components are fed into two separate deep networks. Each stream performs action recognition on its own and final predictions are computed as an average of the two outputs.

Along with being highly successful in the video literature, the two-stream approach is a natural fit for injecting our Im2Flow predictions into action recognition with static images. In short, we augment both training and testing images with their respective inferred flow maps, then train the action recognition network with standard procedures. See Sec. 4 for more details.

Why should the inferred motion help action learning with static images? We hypothesize two rationales. First, there is value in elucidating a salient signal for action that can be difficult to learn directly from the images alone. Static images of different action classes can be visually similar, e.g., pull-ups vs. push-ups, brushing teeth vs. applying lipstick. The motion implied for people and objects in the static images can help better distinguish subtle differences among such actions. This parallels what is currently observed in the literature with real optical flow: presenting an action recognition network with explicit optical flow maps is much stronger than simply presenting the two source

frames—even though the optical flow engine receives those same two frames [65, 12, 81]. There is value in directing the learner's attention to a complex, high-dimensional signal that is useful but would likely require many orders of magnitude more data to learn simultaneously with the target recognition task. Second, the Im2Flow network leverages a large amount of video to build a motion prior that regularizes the eventual activity learning process. Since action recognition datasets are relatively small w.r.t. the variability with which actions can be performed, a learning algorithm can easily overfit, e.g., to the background of training examples. The domain of inferred motion helps to get rid of elements irrelevant to the action performed in the image, and therefore mitigates overfitting.

# 4. Experiments

Using a total of 7 datasets, we validate our approach for 1) flow prediction accuracy (Sec. 4.1) and 2) action recognition from static images (Sec. 4.2

**Implementation details** We implement our Im2Flow network in Torch and train it with videos from UCF-101 [67] and HMDB-51 [47]. We sample 500,000 frames from UCF-101 and 200,000 frames from HMDB-51 as our training data. We use minibatch SGD with a batch size of 32 and apply Adam solver [42]. We train with random horizontal flips and randomly cropped windows as data augmentation. For the motion content loss network, we use the activation maps after the second residual block of ResNet18 and we set $\lambda = 0.02$ in Equation (2). All our action recognition experiments are implemented in Caffe [38]. For action recognition, we use AlexNet [45] with batch normalization [30] as the base architecture for each stream. We fuse the two streams' softmax prediction scores using the optimal weight determined on a validation set.

## 4.1. Flow Prediction

First, we directly evaluate Im2Flow's dense optical flow prediction. Here we use three datasets: UCF-101 [67], HMDB-51 [47], and Weizmann [24]. For UCF-101 and HMDB-51, we hold out 10 videos from each class as test data and the rest as candidate training data; for Weizmann, we hold out the frames of *shahar* as the test set. We compare with the following methods:

- **Walker *et al*. [77]**: Existing CNN-based method that classifies each region in the image to a quantized optical flow vector. We use their publicly available model[2], which is trained on the whole UCF-101 dataset.
- **Pintea *et al*. [58]**: Existing structured random forest regression approach. We use their publicly available code[3] and train a model with default parameters.
- **Nearest Neighbor**: Baseline that uses the pool5 features from a pre-trained AlexNet to retrieve the nearest training image, then adopts the ground-truth flow of that image. Its training pool consists of the same frames that train Im2Flow. This baseline is inspired by the method of Yuen & Torralba [87], which identifies likely future events using nearest neighbor.

**Evaluation metrics** We convert Im2Flow's outputs back to dense optical flow to compare against the "ground truth" flow, which is computed from video with [53]. We employ a suite of metrics, following prior work in this area [58, 77]: End-Point-Error (EPE), Direction Similarity (DS), and Orientation Similarity (OS) (see Supp. for details). Apart from evaluating over all the pixels in the whole image, we also evaluate over masks on the 1) Canny edges, which approximates measuring the error of moving pixels in simple scenes [58, 77], and 2) foreground (FG) regions (computed with [34]), which often correspond to the moving objects.

**Results** Table 1 shows the results on UCF-101 (see Supp. for similar results on HMDB and Weizmann). Our method outperforms both prior work and the Nearest Neighbor baseline consistently by a large margin on all datasets across all metrics. This result shows the effectiveness of the proposed motion encoding and Im2Flow network.

Fig. 3 shows qualitative results. Our Im2Flow network can predict motion in a variety of contexts. The structured random forest approach by Pintea *et al*. [58] makes reasonable predictions on Weizmann, but struggles on more complicated datasets such as HMDB-51. The classification approach by Walker *et al*. [77] predicts plausible motions in many cases, but the predictions are inherently coarse and usually only depict the general trend of motion of the objects in the scene. Our Im2Flow network makes more reliable and fine-grained predictions. For example, in the baby crawling case, while [77] can only predict that the baby is going to move leftwards, our model predicts motion at various body parts of the baby. Similarly, in the example of a boy playing the violin, our model makes reasonable predictions across various parts of the image. Moreover, aside from human motions, our model can also predict scene motions, such as the falling waves in the ocean. However, our motion prediction model is far from perfect. It can fail especially when the motion present in the static image is subtle or the background is too diverse, as shown in the failure cases (last row) in Fig. 3.

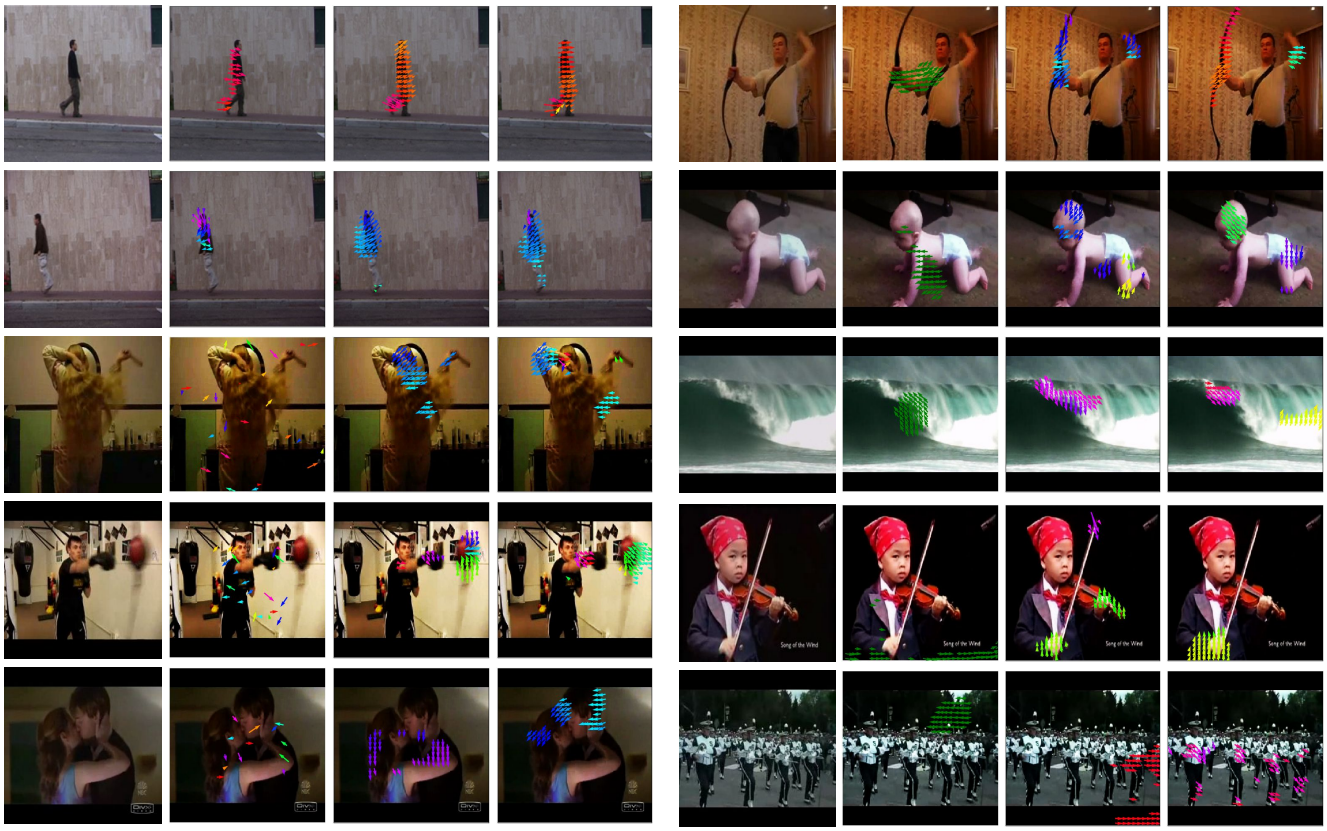
With the ability to anticipate flow, Im2Flow can infer the *motion potential* of a novel image—that is, the strength of movement and activity that is poised to happen. Given an image, we compute its motion potential score by inferring flow, then normalizing the average magnitude by the area of the foreground (obtained using [34]) to avoid a bias to large objects. Fig. 4 shows static images our system rates

[2] https://github.com/puffin444/optical_flow_prediction

[3] https://github.com/SilviaLauraPintea/DejaVu

| UCF-101 | EPE ↓ | EPE-Canny | EPE-FG | DS ↑ | DS-Canny | DS-FG | OS ↑ | OS-Canny | OS-FG |
|---|---|---|---|---|---|---|---|---|---|
| Pintea *et al*. [58] | 2.401 | 2.699 | 3.233 | -0.001 | -0.002 | -0.005 | 0.513 | 0.544 | 0.555 |
| Walker *et al*. [77] | 2.391 | 2.696 | 3.139 | 0.003 | 0.001 | 0.014 | 0.661 | 0.673 | 0.662 |
| Nearest Neighbor | 3.123 | 3.234 | 3.998 | -0.002 | -0.001 | -0.023 | 0.652 | 0.651 | 0.659 |
| Ours | **2.210** | **2.533** | **2.936** | **0.143** | **0.135** | **0.137** | **0.699** | **0.692** | **0.696** |

Table 1. Quantitative results of dense optical flow prediction on UCF-101. ↓ lower better, ↑ higher better. Across all measures, our method outperforms all baseline methods by a large margin. See Supp. for similar results on HMDB-51 and Weizmann datasets.



(a) Input Image (b) Pintea *et al*. [58] (c) Ours (d) Ground-truth (a) Input Image (b) Walker *et al*. [77] (c) Ours (d) Ground-truth

Figure 3. Examples of dense optical flow prediction (best viewed in color). The Pintea *et al*. [58] approach makes reasonable predictions on Weizmann (top left), but suffers on more complex datasets like HMDB-51 (bottom left). The Walker *et al*. [77] approach often captures the general trend of motion, but the predicted motion is coarse. Our Im2Flow network accurately predicts motion that is much more fine-grained in various contexts. The last row shows two failure cases. We use the color coding on the right for flow visualization.

as having the greatest/least motion potential. Motion potential offers a high-level view of a scene's activity, identifying images that are most suggestive of coming events.

### 4.2. Action Recognition

Having demonstrated the accuracy of our flow estimates, we now examine the impact on static-image action recognition. For these experiments, we use seven total datasets: three static-image datasets we construct from existing video datasets, three existing static-image benchmarks, and one dynamic scene dataset.

The three constructed datasets draw on videos from UCF-101 [67], HMDB-51 [47], and Penn Action [88]. For each, we construct static-image datasets by taking the standard train-test splits and extracting the center frame in each video. This yields train/test sets of 9,537/3,783 (UCF-Static), 3,570/1,530 (HMDB-Static), and 1,258/1,068 (Penn Actions) images, respectively. Since they originate from video, these datasets allow us to compute the actual flow, and thereby place an upper bound on its role in static-image action recognition.

The three static-image action benchmarks are Willow [6], Stanford10 [84, 4], and PASCAL2012 Ac-

| | | UCF-static | HMDB-static | PennAction | Willow | Stanford10 | PASCAL2012 |
|---|---|---|---|---|---|---|---|
| | Appearance Stream | 63.6 | 35.1 | 73.1 | 65.1 | 81.3 | 65.0 |
| Motion Stream | Motion Stream (Walker *et al*. [77]) | *14.3 | 4.96 | 21.2 | 18.8 | 19.0 | 15.9 |
| | Motion Stream (Ours-UCF) | - | **13.9** | 51.0 | 35.7 | 46.4 | 32.5 |
| | Motion Stream (Ours-HMDB) | **24.1** | - | 42.4 | 30.6 | 42.2 | 30.1 |
| | Motion Stream (Ours-UCF+HMDB) | - | - | **51.1** | **35.9** | **48.4** | **32.7** |
| | → Motion Stream (Ground-truth Motion) | 38.7 | 20.0 | 52.4 | - | - | - |
| Two-Stream | Appearance + Appearance | 64.0 | 35.5 | 73.4 | 65.8 | 81.3 | 65.1 |
| | Appearance + Motion (Walker *et al*. [77]) | *64.5 | 35.9 | 73.1 | 65.9 | 81.5 | 65.0 |
| | Appearance + Motion (Ours-UCF) | - | **37.1** | 74.5 | 67.4 | 82.1 | 66.0 |
| | Appearance + Motion (Ours-HMDB) | **65.5** | - | 74.3 | 67.1 | 81.9 | 65.6 |
| | Appearance + Motion (Ours-UCF+HMDB) | - | - | **74.5** | **67.5** | **82.3** | **66.1** |
| | → Appearance + Motion (Ground-truth Motion) | 68.1 | 39.5 | 77.4 | - | - | - |

Table 2. Accuracy results (in %) on static-image action recognition datasets. Note that for UCF/HMDB-static and PennAction, the methods train from the static center frames of the videos. Dashes indicate results that would require train/test overlaps, and hence are omitted for our approach. → indicates the performance upper bound by using ground-truth motion. *The model provided by Walker *et al*. [77] is trained on the whole UCF-101 dataset, hence it may have some mild advantage due to overlap with the test data in the starred case. The inferred motion from our Im2Flow framework performs much better than Walker *et al*. (Motion Stream—Ours vs. Walker *et al*.) for recognition. Injecting our inferred motion into a standard two-stream network achieves significant gains compared to the one-stream counterpart (Two-Stream Ours vs. Appearance Stream).


Figure 4. Examples of static images with the greatest/least motion potential determined by our Im2Flow framework.

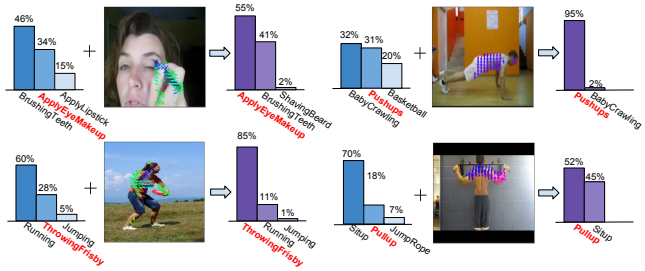



Figure 5. Examples of how the inferred motion can help static-image action recognition. For each example, the left shows the classification results of the appearance stream, and the right shows the two-stream results after incorporating the inferred motion.

tions [10]. Willow has 7 action classes, with 427 training images and 484 testing images. Stanford10 is a subset of Stanford40 [84] generated in [4]. It contains the 10 action classes most related to human action (1,000/1,672 train/test images), as opposed to being characterized by the objects that appear in the scene. For PASCAL2012 Actions, we use bounding-box-cropped images from the standard train/val sets, leading to a train/test set of 2,645/2,658 images.

In all results, we train Im2Flow with UCF-101 (Ours-UCF), HMDB-51 (Ours-HMDB), or their combination (Ours-UCF+HMDB). We then use the trained networks to predict flow images for all the static-image datasets except for the dataset the network is trained on. Thus, we test whether motion learned from disjoint videos/labels can transfer to static images from another domain.

We compare to the following baselines:

- **Appearance Stream:** The recognition network is trained only on the original static images, representing the status quo in static-image action recognition.
- **Motion Stream (Ground-truth):** The recognition network uses "ground-truth" optical flow computed from the video frames. This baseline is only possible for UCF-static, HMDB-static, and PennAction.
- **Motion Stream (Walker *et al*. [77]):** We use the publicly available optical flow prediction model of [77] to generate the motion stream's inputs.
- **Appearance + Appearance:** A standard model that ensembles two separately trained appearance streams to give more robust predictions.

We stress that all recognition baselines employ the same two-stream architecture, differing only in the source of the second stream.

Table 2 shows the action recognition results. In the top part of the table, we show the performance of using a single stream. Although the model of Walker *et al*. [77] can predict coarse optical flow successfully in many cases as shown in Fig. 3, the predicted coarse motion works poorly for recognition. The inferred motion from our Im2Flow framework performs much better, even as well as the ground-truth motion in some cases. The bottom part shows the two-stream performance after combining the appearance and motion

streams. Across all six datasets, we obtain large gains (1-6% relative gain for ours vs. appearance stream) by inferring motion. Although the test cases are from different domains than that which trained our flow network, our approach generalizes well due to the motion prior transferred from unlabeled video to static images. Additionally, we use our Im2Flow network to predict motion (independently) for each UCF-101 frame and then use the method from [81] to train the temporal stream using the predicted motion images. With BN-Inception as the base network, we achieve 90.5% accuracy on UCF, comparable to the SoTA on *video* and further suggesting the power of using predicted motion.

Fig. 5 shows some qualitative results from various datasets to illustrate how the inferred motion can help recognition. While a classifier solely based on appearance can be confused by actions appearing in similar contexts, the inferred motion provides cues about the fine-grained differences among these actions to help recognition. For instance, the first image shows a woman applying eye makeup. However, brushing teeth, applying eye makeup, and applying lipstick are all visually similar. Showing the hand movements of the woman guides the classifier to make the correct prediction. Moreover, our model can even make reasonable predictions for actions that do not appear in the training set, e.g., throwing frisby is a novel action in Stanford 10 dataset, but the inferred motion can still help recognition. See Supp. for more examples.

Finally, we use YUP++ Dynamic Scenes [11] to explore how inferred motion may benefit dynamic scene recognition. Table 3 shows the results. We include this scenario since motion is also indicative in many dynamic natural scenes, e.g, waterfall, falling trees, rushing river. Given a static image of a dynamic scene, hallucinating motion from the scene may also help recognition. We use 90% of the dataset (using the standard 10-90 split) to train our Im2Flow prediction network. From the remaining 10%, we construct the "static-YUP++" dataset for *static-image* dynamic scene recognition. Specifically, we use 2/3 (from the 10% reserved videos) as training data and 1/3 as test data. Once again, with the inferred motion, the recognition accuracy improves by a large margin (78.2% vs. 74.3%) compared to using only static images alone.

**Comparison to alternative recognition models** The results above are all apples-to-apples, in that the only moving part is whether and how implied flow is injected into a two-stream recognition architecture. Next we briefly place our action recognition results on an absolute scale against reported results in the literature.

On Stanford10, the method of [4] uses unlabeled video as a means to generate synthetic training examples in pose space for the low-shot training regime. With 250 training images per class, their method yields 50.2 mAP, whereas our method achieves 74.9 mAP. However, note that our method also benefits from using a deep learning approach.

| | Accuracy | mAP |
|---|---|---|
| Appearance | 74.3 | 79.3 |
| Ground-truth Motion | 55.5 | 62.0 |
| Inferred Motion | 30.0 | 37.0 |
| Appearance + Appearance | 75.2 | 79.8 |
| Appearance + Inferred Motion | **78.2** | **82.3** |
| Appearance + Ground-truth Motion | 79.6 | 83.6 |

Table 3. Static-image dynamic scene recognition results (in %) on the static-YUP++ dataset [11]. The inferred scene motion improves the recognition accuracy by a large margin.

| | mAP (%) |
|---|---|
| Delaitre *et al*. [6] | 59.6 |
| Sharma *et al*. [64] | 67.6 |
| Khan *et al*. [41] | 68.0 |
| Zhang *et al*. [89] | 77.0 |
| Liang *et al*. [51] | 80.4 |
| Mettes *et al*. [57] | 81.7 |
| Ours (AlexNet as base network) | 74.0 |
| Ours (VGG-16 as base network) | 87.2 |
| Ours (ResNet-50 as base network) | **90.5** |

Table 4. Comparison to other recognition models on Willow [6].

For Willow, Table 4 compares our results to several state-of-the-art methods. We attempt three variants of our approach using AlexNet, VGG-16, and ResNet-50 as the base network, respectively. Our approach combines appearance and the inferred motion, and performs well compared to all baselines. Of note, our model with VGG-16 as the base network significantly outperforms Zhang *et al*. [89], who also use VGG-16. Without using separate body part and/or object detectors as in [51, 57], our end-to-end recognition model compares favorably.

## 5. Conclusion

We presented an approach to hallucinate the motion implied by a single snapshot and then use it as an auxiliary cue for static-image action recognition. Our Im2Flow framework achieves state-of-the-art performance on optical flow prediction from an individual image. Moreover, using a standard two-stream network, it enhances recognition of actions and dynamic scenes by a good margin. As future work, we plan to explore hierarchical representations to encode the temporal evolution of multiple video frames.

**Acknowledgements:** This research was supported in part by an ONR PECASE Award N00014-15-1-2291 and an IBM Faculty Award and IBM Open Collaboration Award. We thank Suyog Jain, Chao-Yeh Chen, Aron Yu, Yu-Chuan Su, Tushar Nagarajan and Zhengpei Yang for helpful input on experiments or reading paper drafts, and also gratefully acknowledge a GPU donation from Facebook.

## References


[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011. 3

[2] Y.-W. Chao, Z. Wang, Y. He, J. Wang, and J. Deng. Hico: A benchmark for recognizing human-object interactions in images. In *ICCV*, 2015. 2

[3] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng. Forecasting human dynamics from static images. In *CVPR*, 2017. 2

[4] C.-Y. Chen and K. Grauman. Watching unlabeled video helps learn new human actions from very few labeled snapshots. In *CVPR*, 2013. 2, 6, 7, 8

[5] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition. In *ICCV*, 2015. 4

[6] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *BMVC*, 2010. 6, 8

[7] V. Delaitre, J. Sivic, and I. Laptev. Learning person-object interactions for action recognition in still images. In *NIPS*, 2011. 2

[8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2, 3, 4

[9] A. A. Efros, A. C. Berg, G. Mori, J. Malik, et al. Recognizing action at a distance. In *ICCV*, 2003. 1

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 6

[11] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Temporal residual networks for dynamic scene recognition. In *CVPR*, 2017. 8

[12] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 1, 2, 3, 4, 5

[13] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015. 2

[14] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*, 2016. 2

[15] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 3

[16] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*, 2015. 2

[17] D. F. Fouhey and C. L. Zitnick. Predicting object dynamics in scenes. In *CVPR*, 2014. 2

[18] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *ICCV*, 2015. 2

[19] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell. ActionVLAD: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017. 1, 2, 4

[20] G. Gkioxari, R. Girshick, and J. Malik. Contextual action recognition with r* cnn. In *CVPR*, 2015. 2

[21] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015. 4

[22] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends in neurosciences*, 1992. 4

[23] 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

[24] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *TPAMI*, 2007. 5

[25] G. Guo and A. Lai. A survey on still image based human action recognition. *Pattern Recognition*, 2014. 1, 2

[26] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *TPAMI*, 2009. 2

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[28] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001. 2

[29] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 3

[30] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 3, 5

[31] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 3, 4

[32] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis. Representing videos using mid-level discriminative patches. In *CVPR*, 2013. 2

[33] S. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *CVPR*, 2017. 3

[34] S. Jain, B. Xiong, and K. Grauman. Pixel objectness. *arXiv preprint arXiv:1701.05349*, 2017. 5

[35] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007. 1

[36] D. Ji, J. Kwon, M. McFarland, and S. Savarese. Deep view morphing. In *CVPR*, 2017. 2

[37] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 2013. 1

[38] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[39] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 3

[40] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1, 2

[41] F. S. Khan, J. Van De Weijer, A. D. Bagdanov, and M. Felsberg. Scale coding bag-of-words for action recognition. In *ICPR*, 2014. 8

[42] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[43] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012. 2

[44] Z. Kourtzi and N. Kanwisher. Activation in human mt/mst by static images with implied motion. *Journal of cognitive neuroscience*, 2000. 1

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3, 5

[46] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool. Fast optical flow using dense inverse search. In *ECCV*, 2016. 3

[47] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 5, 6

[48] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003. 1, 2

[49] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 1

[50] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *CVPR*, 2017. 2

[51] Z. Liang, X. Wang, R. Huang, and L. Lin. An expressive deep model for human action parsing from a single image. In *ICME*, 2014. 8

[52] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009. 1

[53] C. Liu et al. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 3, 5

[54] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017. 2

[55] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR*, 2011. 2

[56] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 2

[57] P. Mettes, J. C. van Gemert, and C. G. Snoek. No spare parts: Sharing part detectors for image categorization. *CVIU*, 2016. 8

[58] S. L. Pintea, J. C. van Gemert, and A. W. Smeulders. Déja vu. In *ECCV*, 2014. 2, 5, 6

[59] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *TPAMI*, 2012. 2

[60] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014. 2

[61] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012. 2

[62] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 3

[63] F. Sener, C. Bas, and N. Ikizler-Cinbis. On recognizing actions in still images via multiple features. In *ECCV*, 2012. 2

[64] G. Sharma, F. Jurie, and C. Schmid. Expanded parts model for human attribute and action recognition in still images. In *CVPR*, 2013. 8

[65] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2, 3, 4, 5

[66] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3

[67] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 3, 5, 6

[68] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 2, 4

[69] C. Thurau and V. Hlavác. Pose primitive based human action recognition in videos or still images. In *CVPR*, 2008. 2

[70] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *PAMI*, 2017. 1, 2, 3

[71] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017. 2

[72] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. In *CVPR*, 2016. 2

[73] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016. 2

[74] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *CVPR*, 2017. 2

[75] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 2

[76] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *CVPR*, 2014. 2

[77] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *ICCV*, 2015. 2, 3, 5, 6, 7

[78] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 1, 2

[79] H. Wang and C. Schmid. Action recognition with improved trajectories. In *CVPR*, 2013. 1, 2

[80] L. Wang, Y. Qiao, and X. Tang. Motionlets: Mid-level 3d parts for human motion recognition. In *CVPR*, 2013. 2

[81] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks for action recognition in videos. *TPAMI*, 2017. 1, 3, 4, 5, 8

[82] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008. 1, 2

[83] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016. 2

[84] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011. 2, 6, 7

[85] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 3

[86] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 1, 2, 3, 4

[87] J. Yuen and A. Torralba. A data-driven approach for event prediction. In *ECCV*, 2010. 2, 5

[88] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013. 6

[89] Y. Zhang, L. Cheng, J. Wu, J. Cai, M. N. Do, and J. Lu. Action recognition in still images with minimum annotation efforts. *IEEE Transactions on Image Processing*, 2016. 8

[90] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2, 4