# Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks

Pavlo Molchanov Xiaodong Yang Shalini Gupta Kihwan Kim Stephen Tyree Jan Kautz
NVIDIA
{pmolchanov,xiaodongy,shalinig,kihwank,styree,jkautz}@nvidia.com

## Abstract

*Automatic detection and classification of dynamic hand gestures in real-world systems intended for human computer interaction is challenging as: 1) there is a large diversity in how people perform gestures, making detection and classification difficult; 2) the system must work online in order to avoid noticeable lag between performing a gesture and its classification; in fact, a negative lag (classification before the gesture is finished) is desirable, as feedback to the user can then be truly instantaneous. In this paper, we address these challenges with a recurrent three-dimensional convolutional neural network that performs simultaneous detection and classification of dynamic hand gestures from multi-modal data. We employ connectionist temporal classification to train the network to predict class labels from in-progress gestures in unsegmented input streams. In order to validate our method, we introduce a new challenging multi-modal dynamic hand gesture dataset captured with depth, color and stereo-IR sensors. On this challenging dataset, our gesture recognition system achieves an accuracy of* $83.8\%$*, outperforms competing state-of-the-art algorithms, and approaches human accuracy of* $88.4\%$*. Moreover, our method achieves state-of-the-art performance on SKIG and ChaLearn2014 benchmarks.*

## 1. Introduction

Hand gestures and gesticulations are a common form of human communication. It is therefore natural for humans to use this form of communication to interact with machines as well. For instance, touch-less human computer interfaces can improve comfort and safety in vehicles. Computer vision systems are useful tools in designing such interfaces. Recent work using deep convolutional neural networks (CNN) with video sequences has significantly advanced the accuracy of dynamic hand gesture [22, 23, 25] and action [13, 34, 37] recognition. CNNs are also useful for combining multi-modal data inputs [23, 25], a technique which has proved useful for gesture recognition in challenging lighting conditions [23, 27].

However, real-world systems for dynamic hand gesture recognition present numerous open challenges. First, these systems receive continuous streams of unprocessed visual data, where gestures from known classes must be simultaneously detected and classified. Most prior work, e.g., [21, 23, 25, 27], regards gesture segmentation and classification separately. Two classifiers, a detection classifier to distinguish between *gesture* and *no gesture* and a recognition classifier to identify the specific gesture type, are often trained separately and applied in sequence to the input data streams. There are two reasons for this: (1) to compensate for variability in the duration of the gesture and (2) to reduce noise due to unknown hand motions in the *no gesture* class. However, this limits the system's accuracy to that of the upstream detection classifier. Additionally, since both problems are highly interdependent, it is advantageous to address them jointly. A similar synergy was shown to be useful for joint face detection and pose estimation [28].

Second, dynamic hand gestures generally contain three temporally overlapping phases: preparation, nucleus, and retraction [8, 14], of which the nucleus is most discriminative. The other two phases can be quite similar for different gesture classes and hence less useful or even detrimental to accurate classification. This motivates designing classifiers which rely primarily on the nucleus phase.

Finally, humans are acutely perceptive of the response time of user interfaces, with lags greater than $100$ ms perceived as annoying [3, 20]. This presents the challenge of detecting and classifying gestures immediately upon or before their completion to provide rapid feedback.

In this paper, we present an algorithm for joint segmentation and classification of dynamic hand gestures from continuous depth, color and stereo-IR data streams. Building on the recent success of CNN classifiers for gesture recognition, we propose a network that employs a recurrent three dimensional (3D)-CNN with connectionist temporal classification (CTC) [10]. CTC enables gesture classification to be based on the nucleus phase of the gesture without requiring explicit pre-segmentation. Furthermore, our network

addresses the challenge of early detection of gestures, resulting in zero or negative lag, which is a crucial element for responsive user interfaces. We present a new multi-modal hand gesture dataset[1] with 25 classes for comparing our algorithm against state-of-the-art methods and human subject performance.

## 2. Related Work

Many hand-crafted spatio-temporal features for effective video analysis have been introduced in the area of gesture and action recognition [33, 36, 39]. They typically capture shape, appearance, and motion cues via image gradients and optical flow. Ohn-Bar and Trivedi [27] evaluate several global features for automotive gesture recognition. A number of video classification systems successfully employ improved dense trajectories [39] and Fisher vector [30] representations, which are widely regarded as state-of-the-art local features and aggregation techniques for video analysis. Features for depth sensors are usually designed according to the specific characteristics of the depth data. For instance, random occupancy patterns [40] utilize point clouds and super normal vectors [42] employ surface normals.

In contrast to hand-crafted features, there is a growing trend toward feature representations learned by deep neutral networks. Neverova *et al*. [25] employ CNNs to combine color and depth data from hand regions and upper-body skeletons to recognize sign language gestures. Molchanov *et al*. [22, 23] apply a 3D-CNN on the whole video sequence and introduce space-time video augmentation techniques to avoid overfitting. In the context of action recognition, Simonyan and Zisserman [34] propose separate CNNs for the spatial and temporal streams that are late-fused and that explicitly use optical flow. Tran *et al*. [37] employ a 3D-CNN to analyze a series of short video clips and average the network's responses for all clips. Most previous methods either employ pre-segmented video sequences or treat detection and classification as separate problems.

To the best of our knowledge, none of the previous methods for hand gesture recognition address the problem of early gesture recognition to achieve the zero or negative lag necessary for designing effective gesture interfaces. Early detection techniques have been proposed for classifying facial expressions and articulated body motion [12, 32], as well as for predicting future events based on incoming video streams [15, 16]. The predicted motions in many of these methods are aided by the appearance of their environments (i.e., road or parking lot)—something we cannot rely on for gesture recognition. Recently, connectionist temporal classification has been shown to be effective for classification of unsegmented handwriting and speech [9, 10]. We demonstrate the applicability of CTC for gesture recognition from unsegmented video steams.

[1] https://research.nvidia.com/publication/online-detection-and-classification-dynamic-hand-gestures-recurrent-3d-convolutional

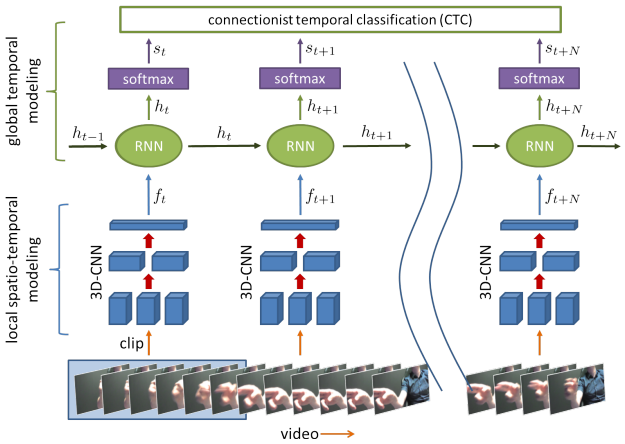


Figure 1: Classification of dynamic gestures with R3DCNN. A gesture video is presented in the form of short clips $C_t$ to a 3D-CNN for extracting local spatial-temporal features, $\mathbf{f}_t$. These features are input to a recurrent network, which aggregates transitions across several clips. The recurrent network has a hidden state $\mathbf{h}_{t-1}$, which is computed from the previous clips. The updated hidden state for the current clip, $\mathbf{h}_t$, is input into a softmax layer to estimate class-conditional probabilities, $\mathbf{s}_t$ of the various gestures. During training, CTC is used as the cost function.

## 3. Method

In this section, we describe the architecture and training of our algorithm for multi-modal dynamic hand gesture detection and classification.

### 3.1. Network Architecture

We propose a recurrent 3D convolutional neural network (R3DCNN) for dynamic hand gesture recognition, illustrated in Fig. 1. The architecture consists of a deep 3D-CNN for spatio-temporal feature extraction, a recurrent layer for global temporal modeling, and a softmax layer for predicting class-conditional gesture probabilities.

We begin by formalizing the operations performed by the network. We define a video clip as a volume $C_t \in \mathbb{R}^{k\times\ell\times c\times m}$ of $m \geq 1$ sequential frames with $c$ channels of size $k \times \ell$ pixels ending at time $t$. Each clip is transformed into a feature representation $\mathbf{f}_t$ by a 3D-CNN $\mathcal{F}$:

$$\mathcal{F} : \mathbb{R}^{k\times\ell\times c\times m} \rightarrow \mathbb{R}^q, \text{where } \mathbf{f}_t = \mathcal{F}(C_t),$$

by applying spatio-temporal filters to the clip. A recurrent layer computes a hidden state vector $\mathbf{h}_t \in \mathbb{R}^d$ as a function of the hidden state after the previous clip $\mathbf{h}_{t-1}$ and the feature representation of the current clip $\mathbf{f}_t$:

$$\mathbf{h}_t = \mathcal{R}(W_{in}\mathbf{f}_t + W_h\mathbf{h}_{t-1}),$$

with weight matrices $W_{in} \in \mathbb{R}^{d\times q}$ and $W_h \in \mathbb{R}^{d\times d}$, and truncated rectified linear unit $\mathcal{R} : \mathbb{R}^d \rightarrow \mathbb{R}^d$, $\mathcal{R}(x) = \min(\max(0, x), 4)$ to limit gradient explosion [29] during training. Finally, a softmax layer transforms the hidden state vector $\mathbf{h}_t$ into class-conditional probabilities $\mathbf{s}_t$ of $w$ classes:

$$\mathbf{s}_t = \mathcal{S}(W_s\mathbf{h}_t + \mathbf{b}),$$

with weights $W_s \in \mathbb{R}^{w\times d}$, bias $\mathbf{b} \in \mathbb{R}^w$, and a softmax function $\mathcal{S} : \mathbb{R}^w \rightarrow \mathbb{R}^w_{[0,1]}$, where $[\mathcal{S}(\mathbf{x})]_i = e^{x_i} / \sum_k e^{x_k}$.

We perform classification by splitting the entire video $\mathcal{V}$ into $T$ clips of length $m$ and computing the set of class-conditional probabilities $S = \{\mathbf{s}_0, \mathbf{s}_1, ..., \mathbf{s}_{T-1}\}$ for each individual clip. For offline gesture classification, we average the probabilities of all the clips belonging to a pre-segmented gesture $\mathbf{s}^{\text{avg}} = 1/T \sum_{\mathbf{s}\in S} \mathbf{s}$, and the predicted class is $\hat{y} = \text{argmax}_i\,([\mathbf{s}^{\text{avg}}]_i)$, across all gesture classes $i$. When predicting online with unsegmented streams, we consider only clip-wise probabilities $\mathbf{s}_t$.

We combine multiple modalities by avereging the class-conditional probabilities estimated by the modality-specific networks. During online operation, we average probabilities across modalities for the current clip only. As an alternative to the softmax layer, we additionally consider computing the final classification score with a support vector machine (SVM) [6] classifier operating on features $\mathbf{f}_t$ or $\mathbf{h}_t$ extracted by the R3DCNN. We average the features across video clips and normalize by their $\ell_2$-norms to form a single representation for the entire video.

### 3.2. Training

Let $\mathcal{X} = \{\mathcal{V}_0, \mathcal{V}_1, ..., \mathcal{V}_{P-1}\}$ be a mini-batch of training examples in the form of weakly-segmented gesture videos $\mathcal{V}_i$.[2] Each video consists of $T$ clips, making $\mathcal{X}$ a set of $N = T\cdot P$ clips. Class labels $y_i$ are drawn from the alphabet $\mathcal{A}$ to form a vector of class labels $\mathbf{y}$ with size $|\mathbf{y}| = P$.

**Pre-training the 3D-CNN.** We initialize the 3D-CNN with the C3D network [37] trained on the large-scale Sport-1M [13] human action recognition dataset. The network has 8 convolutional layers of 3×3×3 filters and 2 fully-connected layers trained on 16-frame clips. We append a softmax prediction layer to the last fully-connected layer and fine-tune by back-propagation with negative log-likelihood to predict gestures classes from individual clips $C_i$.

**Training the full model.** After fine-tuning the 3D-CNN, we train the entire R3DCNN with back-propagation-through-time (BPTT) [41]. BPTT is equivalent to unrolling the recurrent layers, transforming them into a multi-layer feed-forward network, applying standard gradient-based back-propagation, and averaging the gradients to consolidate updates to weights duplicated by unrolling.

[2]Weakly-segmented videos contain the preparation, nucleus, and retraction phases and frames from the *no gesture* class.

We consider two training cost functions: negative log-likelihood for the entire video and connectionist temporal classification (CTC) for online sequences. The negative log-likelihood function for a mini-batch of videos is:

$$\mathcal{L}_v = -\frac{1}{P}\sum_{i=0}^{P-1} \log\big(p(y_i|\mathcal{V}_i)\big),$$

where $p(y_i|\mathcal{V}_i) = [\mathbf{s}^{\text{avg}}]_{y_i}$ is the probability of gesture label $y_i$ given gesture video $\mathcal{V}_i$ as predicted by R3DCNN.

**Connectionist temporal classification.** CTC is a cost function designed for sequence prediction in unsegmented or weakly segmented input streams [9, 10]. CTC is applied in this work to identify and correctly label the nucleus of the gesture, while assigning the *no gesture* class to the remaining clips, addressing the alignment of class labels to particular clips in the video. In this work we consider only the CTC forward algorithm.

We extend the dictionary of existing gestures with a *no gesture* class: $\mathcal{A}' = \mathcal{A} \cup \{\textit{no gesture}\}$. Consequently, the softmax layer outputs a class-conditional probability for this additional *no gesture* class. Instead of averaging predictions across clips in a pre-segmented gesture, the network computes the probability of observing a particular gesture (or *no gesture*) $k$ at time $t$ in an input sequence $\mathcal{X}$: $p(k, t|\mathcal{X}) = \mathbf{s}_t^k \;\; \forall t \in [0, N)$.

We define a path $\pi$ as a possible mapping of the input sequence $\mathcal{X}$ into a sequence of class labels $\mathbf{y}$. The probability of observing path $\pi$ is $p(\pi|\mathcal{X}) = \prod_t \mathbf{s}_t^{\pi_t}$, where $\pi_t$ is the class label predicted at time $t$ in path $\pi$.

Paths are mapped into a sequence of event labels $\mathbf{y}$ by operator $\mathcal{B}$ as $\mathbf{y} = \mathcal{B}(\pi)$, condensing repeated class labels and removing *no gesture* labels, e.g., $\mathcal{B}([-, 1, 2, -, -]) = \mathcal{B}([1, 1, -, 2, -]) = [1, 2]$, where $1, 2$ are actual gesture classes and "$-$" is *no gesture*. Under $\mathcal{B}$, many paths $\pi$ result in the same event sequence $\mathbf{y}$. The probability of observing a particular sequence $\mathbf{y}$ given an input sequence $\mathcal{X}$ is the sum of the conditional probabilities of all paths $\pi$ mapping to that sequence, $\mathcal{B}^{-1}(\mathbf{y}) = \{\pi : \mathcal{B}(\pi) = \mathbf{y}\}$:

$$p(\mathbf{y}|\mathcal{X}) = \sum_{\pi\in\mathcal{B}^{-1}(\mathbf{y})} p(\pi|\mathcal{X}).$$

Computation of $p(\mathbf{y}|\mathcal{X})$ is simplified by dynamic programing. First, we create an assistant vector $\dot{\mathbf{y}}$ by adding a *no gesture* label before and after each gesture clip in $\mathbf{y}$, so that $\dot{\mathbf{y}}$ contains $|\dot{\mathbf{y}}| = P' = 2P + 1$ labels. Then, we compute a forward variable $\boldsymbol{\alpha} \in \mathbb{R}^{N\times P}$ where $\alpha_t(u)$ is the combined probability of all mappings of events up to clip $t$ and event $u$. The transition function for $\boldsymbol{\alpha}$ is:

$$\alpha_t(u) = \mathbf{s}_t^{\dot{y}_u}\big(\alpha_{t-1}(u) + \alpha_{t-1}(u-1) + \beta_{t-1}(u-2)\big),$$

where

$$\beta_t(u) = \begin{cases} \alpha_t(u), & \text{if } \dot{y}_{u+1} = \textit{no gesture} \text{ and } \dot{y}_u \neq \dot{y}_{u+2} \\ 0, & \text{otherwise} \end{cases}$$

and $\dot{y}_u$ denotes the class label of event $u$. The forward variable is initialized with $\alpha_0(0) = \mathbf{s}_0^{\dot{y}_0}$, the probability of a path beginning with $\dot{y}_0 = \textit{no gesture}$, and $\alpha_0(1) = \mathbf{s}_0^{\dot{y}_1}$, the probability of a path starting with the first actual event $\dot{y}_1$. Since a valid path cannot begin with a later event, we initialize $\alpha_0(i) = 0 \ \forall i > 1$. At each time step $t > 0$, we consider paths in which the event $u$ is currently active (with probability $\mathbf{s}_t^{\dot{y}_u}$) and (1) remains unchanged from the previous time $t-1$ ($\alpha_{t-1}(u)$), (2) changes from *no gesture* to the next actual gesture or vice versa ($\alpha_{t-1}(u-1)$), or (3) transitions from one actual gesture to the next while skipping *no gesture* if the two gestures have distinct labels ($\beta_{t-1}(u-2)$). Finally, any valid path $\pi$ must end at time $N-1$ with the last actual gesture $\dot{y}_{P'-1}$ or with *no gesture* $\dot{y}_{P'}$, hence $p(\mathbf{y}|\mathcal{X}) = \alpha_{N-1}(P'-1) + \alpha_{N-1}(P')$.

Using this computation for $p(\mathbf{y}|\mathcal{X})$, the CTC loss is:

$$\mathcal{L}_{CTC} = -\ln(p(\mathbf{y}|\mathcal{X})),$$

expressed in the log domain [9]. While CTC is used as a training cost function only, it affects the architecture of the network by adding the extra *no gesture* class label. For pre-segmented video classification, we simply remove the *no-gesture* output and renormalize probabilities by the $\ell_1$-norm after modality fusion.

**Learning rule.** To optimize the network parameters $\mathcal{W}$ with respect to either of the loss functions we use stochastic gradient descent (SGD) with a momentum term $\mu = 0.9$. We update each parameter of the network $\theta \in \mathcal{W}$ at every back-propagation step $i$ by:

$$\theta_i = \theta_{i-1} + v_i - \gamma\lambda\theta_{i-1},$$
$$v_i = \mu v_{i-1} - \lambda\mathcal{J}\left(\left\langle \frac{\delta E}{\delta \theta} \right\rangle_{batch}\right),$$

where $\lambda$ is the learning rate, $\langle \frac{\delta E}{\delta \theta} \rangle_{batch}$ is the gradient value of the chosen cost function $E$ with respect to the parameter $\theta$ averaged over the mini-batch, and $\gamma$ is the weight decay parameter. To prevent gradient explosion in the recurrent layers during training, we apply a soft gradient clipping operator $\mathcal{J}(\cdot)$ [29] with a threshold of 10.

**Regularization.** We apply a number of regularization techniques to reduce overfitting. We train with weight decay ($\gamma = 0.5\%$) on all weights in the network. We apply drop-out [11] to the fully-connected layers of the 3D-CNN at a rate of $p = 75\%$, rescaling the remaining activations by a factor of $1/(1-p)$. Additionally, we find that dropping feature maps in the convolutional layers improves generalization in pre-trained networks. For this, we randomly set $10\%$ of the feature maps of each convolutional layer to 0 and rescale the activations of the others neurons accordingly.

**Implementation.** We train our gesture classifier in Theano [2] with cuDNN3 on an NVIDIA DIGITS DevBox with four Titan X GPUs.

We fine-tune the 3D-CNN for 16 epochs with an initial learning rate of $\lambda = 3 \cdot 10^{-3}$, reduced by a factor of 10 after every 4 epochs. Next, we train the R3DCNN end-to-end for an additional 100 epochs with a constant learning rate of $\lambda = 3 \cdot 10^{-4}$. All network parameters without pre-trained initializations are randomly sampled from a zero-mean Gaussian distribution with standard deviation 0.01.

Each video of a weakly-segmented gesture is stored with 80 frames of $120 \times 160$ pixels. We train with frames of size $112 \times 112$ generated by random crops. Videos from the test set are evaluated with the central crop of each frame. To increase variability in the training examples, we apply the following data augmentation steps to each video in addition to cropping: random spatial rotation ($\pm 15°$) and scaling ($\pm 20\%$), temporal scaling ($\pm 20\%$), and jittering ($\pm 3$ frames). The parameters for each augmentation step are drawn from a uniform distribution with a specified range. Since recurrent connections can learn the specific order of gesture videos in the training set, we randomly permute the training gesture videos for each training epoch.

We use CNNs pre-trained on three-channel RGB images. To apply them to one-channel depth or IR images, we sum the convolutional kernels for the three channels of the first layer to obtain one kernel. Similarly, to employ the pre-trained CNN with two-channel inputs (e.g., optical flow), we remove the third channel of each kernel and rescale the first two by a factor of 1.5.

For the 3D-CNN, we find that splitting a gesture into non-overlapping clips of $m = 8$ frames yields the best combination of classification accuracy, computational complexity and prediction latency. To work with clips of size $m = 8$ frames on the C3D network [37] (originally trained with $m = 16$ frames), we remove temporal pooling after the last convolutional layer. Since data transfer and inference on a single 8-frame clip takes less than 30ms on an NVIDIA Titan X, we can predict at a faster rate than clips are accumulated.

## 4. Dataset

Recently, several public dynamic gesture datasets have been introduced [5, 18, 19, 27]. The datasets differ in the complexity of gestures, the number of subjects and gesture classes, and the types of sensors used for data collection. Among them, the Chalearn dataset [5] provides the largest number of subjects and samples, but its 20 gesture classes,

derived from the Italian sign language, are quite different from the set of gestures common for user interfaces. The VIVA challenge dataset [27] provides driver hand gestures performed by a small number of subjects (8) against a plain background and from a single viewpoint.

Given the limitations of existing datasets, to validate our proposed gesture recognition algorithm, we acquired a large dataset of 25 gesture types, each intended for human-computer interfaces and recorded by multiple sensors and viewpoints. We captured continuous data streams, containing a total of 1532 dynamic hand gestures, indoors in a car simulator with both bright and dim artificial lighting (Fig. 2). A total of 20 subjects participated in data collection, some with two recorded sessions and some with partial sessions. Subjects performed gestures with their right hand while observing the simulator's display and controlling the steering wheel with their left hand. An interface on the display prompted subjects to perform each gesture with an audio description and a 5s sample video of the gesture. Gestures were prompted in random order with each type requested 3 times over the course of a full session.

Gestures (Fig. 3) include moving either the hand or two fingers up, down, left or right; clicking with the index finger; beckoning; opening or shaking the hand; showing the index finger, or two or three fingers; pushing the hand up, down, out or in; rotating two fingers clockwise or counter-clockwise; pushing two fingers forward; closing the hand twice; and showing "thumb up" or "OK".

We used the SoftKinetic DS325 sensor to acquire front-view color and depth videos and a top-mounted DUO 3D sensor to record a pair of stereo-IR streams. In addition, we computed dense optical flow [7] from the color stream and the IR disparity map from the IR-stereo pair [4]. We randomly split the data by subject into training (70%) and test (30%) sets, resulting in 1050 training and 482 test videos.

## 5. Results

We analyze the performance of R3DCNN for dynamic gesture recognition and early detection.

### 5.1. Offline Gesture Recognition

**Modality fusion.** We begin by evaluating our proposed R3DCNN classifier for a variety of input modalities contained in our dataset: color (front view), optical flow from color (front view), depth (front view), stereo IR (top view), and IR disparity (top view) (bottom row of Fig. 2). We train a separate network for each modality and, when fusing modalities, average their class-conditional probability vectors.[3] Table 1 contains the accuracy for various combinations of sensor modalities. Observe that fusing any pair of sensors improves individual results. In addition, combining different modalities of the same sensor (e.g., color and optical flow) also improves the accuracy. The best gesture recognition accuracy (83.8%) is observed for the combination of all modalities.

[3] Attempts to learn a parameterized fusion layer resulted in overfitting.

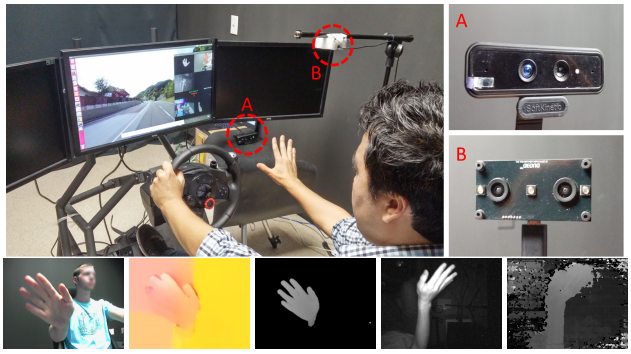


Figure 2: Environment for data collection. (Top) Driving simulator with main monitor displaying simulated driving scenes and a user interface for prompting gestures, (A) a SoftKinetic depth camera (DS325) recording depth and RGB frames, and (B) a DUO 3D camera capturing stereo IR. Both sensors capture $320 \times 240$ pixels at 30 frames per second. (Bottom) Examples of each modality, from left: RGB, optical flow, depth, IR-left, and IR-disparity.

Table 1: Comparison of modalities and their combinations.

| **Sensors** | **Accuracy** | **Combinations** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Depth | 80.3% | | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Optical flow | 77.8% | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Color | 74.1% | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ |
| IR image | 63.5% | ✓ | | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ |
| IR disparity | 57.8% | ✓ | | | ✓ | ✓ | | ✓ | ✓ | | ✓ |
| **Fusion Accuracy** | | 66.2% | 79.3% | 81.5% | 82.0% | 82.0% | 82.4% | 82.9% | 83.2% | 83.4% | **83.8%** |

**Comparisons.** We compare our approach to state-of-the-art methods: HOG+HOG$^2$ [27], improved dense trajectories (iDT) [39], super normal vector (SNV) [42], two-stream CNNs [34], and convolutional 3D (C3D) [37], as well as human labeling accuracy.

To compute the HOG+HOG$^2$ [27] descriptors, we resample the videos to 32 frames and tune the parameters of the SVM classifier via grid search to maximize accuracy. For iDT [39], we densely sample and track interest points at multiple spatial scales, and compute HOG, histogram of optical flow (HOF), and motion boundary histogram (MBH) descriptors from each track. We employ Fisher vectors (FV) [30] to aggregate each type of iDT descriptor using 128 Gaussian components, as well as a spatio-temporal pyramid [17] of FV to encode the space and time information.

Among the CNN-based methods, we compare against the two-stream network for video classification [34], which

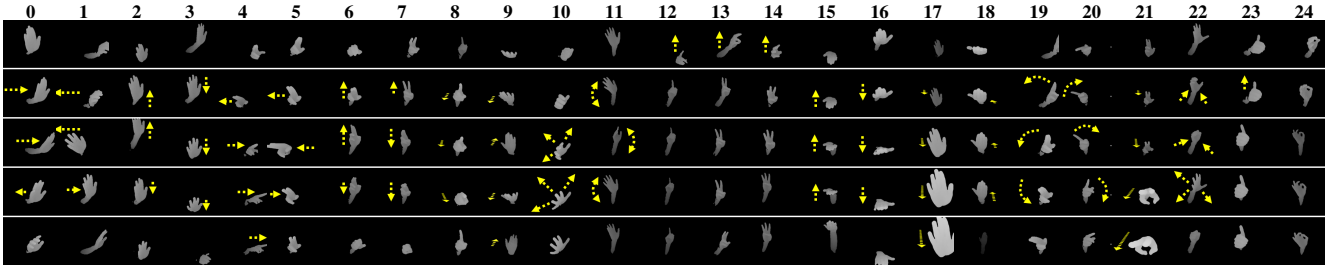


Figure 3: Twenty-five dynamic hand gesture classes. Some gestures were adopted from existing commercial systems [1] or popular datasets [23, 27]. Each column shows a different gesture class ($0-24$). The top and bottom rows show the starting and ending depth frames, respectively, of the nucleus phase for each class. (Note that we did not crop the start and end frames in the actual training and evaluation data.) Yellow arrows indicate the motion of each hand gesture. (A more detailed description of each gesture is available in the supplementary video.)

utilizes the pre-trained VGG-Net [35]. We fine-tune its spatial stream with the color modality and the temporal stream with optical flow, each from our gesture dataset. We also compare against the C3D [37] method, which is trained with the Sports-1M [13] dataset and fine-tuned with the color or depth modalities of our dataset.

Lastly, we evaluate human performance by asking six subjects to label each of the 482 gesture videos in the test set after viewing the corresponding front-view SoftKinetic color video. Prior to the experiment, each subject familiarized themselves with all 25 gesture types. Gestures were presented in random order to each subject for labelling. To be consistent with machine classifiers, human subjects viewed each gesture video only once, but were not restricted in the time allowed to decide each label.

The results of these comparisons are shown in Table 2. Among the individual modalities, the best results are achieved by depth, followed by optical flow and color. This could be because the depth sensor is more robust to indoor lighting change and more easily precludes the noisy background scene, relative to the color sensor. Optical flow explicitly encapsulates motion, which is important to recognize dynamic gestures. Unlike the two-stream network with action classification [34], its accuracy for gesture recognition is not improved by combining the spatial and temporal streams. We conjecture that videos for action classification can be associated with certain static objects or scenes, e.g., sports or ceremonies, which is not the case for dynamic hand gestures. Although C3D captures both shape and motion cues in each clip, the temporal relationship between clips is not considered. Our approach achieves the best performances in each individual modality and significantly outperforms other methods with combined modalities, meanwhile it is still below human accuracy (88.4%).

**Design choices.** We analyze the individual components of our proposed R3DCNN algorithm (Table 3). First, to understand the utility of the 3D-CNN we substitute it with a 2D-CNN initialized with the pre-trained 16-layer VGG-Net [35] and train similarly to the 3D-CNN. We also asses the importance of the recurrent network, the CTC cost function and feature map drop-out. Classification accuracies for these experiments are listed in Table 3. When the recurrent

Table 2: Comparison of our method to the state-of-the-art methods and human predictions with various modalities.

| Method | Modality | Accuracy |
|---|---|---|
| HOG+HOG$^2$ [27] | color | 24.5% |
| Spatial stream CNN [34] | color | 54.6% |
| iDT-HOG [39] | color | 59.1% |
| C3D [37] | color | 69.3% |
| Ours | color | **74.1%** |
| HOG+HOG$^2$ [27] | depth | 36.3% |
| SNV [42] | depth | 70.7% |
| C3D [37] | depth | 78.8% |
| Ours | depth | **80.3%** |
| iDT-HOF [39] | opt flow | 61.8% |
| Temporal stream CNN [34] | opt flow | 68.0% |
| iDT-MBH [39] | opt flow | 76.8% |
| Ours | opt flow | **77.8%** |
| HOG+HOG$^2$ [27] | color + depth | 36.9% |
| Two-stream CNNs [34] | color + opt flow | 65.6% |
| iDT [39] | color + opt flow | 73.4% |
| Ours | all | **83.8%** |
| Human | color | 88.4% |

Table 3: Comparison of 2D-CNN and 3D-CNN trained with different architectures on depth or color data. (CTC$^*$ denotes training without drop-out of feature maps.)

| | Color | | Depth | |
|---|---|---|---|---|
| | **2D-CNN** | **3D-CNN** | **2D-CNN** | **3D-CNN** |
| No RNN | 55.6% | 67.2% | 68.1% | 73.3% |
| RNN | 57.9% | 72.0% | 64.7% | 79.5% |
| CTC | 65.6% | **74.1%** | 69.1% | **80.3%** |
| CTC$^*$ | 59.5% | 66.5% | 67.0% | 75.6% |

Table 4: Accuracy of a linear SVM ($C = 1$) trained on features extracted from different networks and layers (final fully-connected layer `fc` and recurrent layer `rnn`).

| | **Clip-wise C3D** [37] | **R3DCNN** | |
|---|---|---|---|
| **Modality** | `fc` | `fc` | `rnn` |
| Color | 69.3% | 73.0% | **74.1%** |
| Depth | 78.8% | 79.9% | **80.1%** |

network is absent, i.e., "No RNN" in Table 3, the CNN is directly connected to the softmax layer, and the network is trained with a negative log-likelihood cost function. When a recurrent layer with $d = 256$ hidden neurons is present, we train using the negative log-likelihood and CTC cost functions, denoted "RNN" and "CTC" in Table 3, respectively.

We observe consistently superior performance with 3D-CNN versus 2D-CNN for all sensor types and network configurations. This suggests that local motion information extracted by the spatio-temporal kernels of the 3D-CNN is important for dynamic hand gesture recognition. Notice also that adding global temporal modeling via RNN into the classifier generally improves accuracy, and the best accuracy for all sensors is obtained with the CTC cost function, regardless of the type of CNN employed.

Finally, we evaluate the effect of feature map drop-out, which involves randomly setting entire maps to zero while training. This technique has been shown to provide little or no improvement when training from a CNN from scratch [11]. However, when a network pre-trained on a larger dataset with more classes is fine -tuned for a smaller domain with fewer training examples and classes, not all of the original feature maps are likely to exhibit strong activations for the the new inputs. This can lead to overfitting during fine-tuning. The accuracies of the various classifier architectures, trained with and without feature map drop-out are denoted by "CTC" and "CTC*" in Table 3, respectively. They show improved accuracy for all modalities and networks with feature map drop-out, with a greater positive effect for the 3D-CNN.

**Recurrent layer as a regularizer for feature extractors.** Tran *et al*. [37] perform video classification with a linear SVM classifier learned on features extracted from the fully connected layers of the C3D network. Features for each individual clip are averaged to form a single representation for the entire video. In Table 4, we compare the performance of the features extracted from the C3D network fine-tuned on gesture clips with the features from R3DCNN trained with CTC on entire gesture videos. Features extracted from each clip are normalized by the $\ell_2$-norm. Since R3DCNN connects a C3D architecture to a recurrent network, `fc` layer features in both networks are computed by the same architecture, each with weights fine-tuned for the gesture recognition. However, we observe (columns 1-2, Table 4) that following `fc` by a recurrent layer and training on full videos (R3DCNN) improves the accuracy of the extracted features. A plausible explanation is that the recurrent layer help the preceding convolutional network to learn more general features. Moreover, features from the recurrent layer when coupled with an SVM classifier, demonstrate a further improvement in performance (column 3, Table 4).

## 5.2. Early Detection in Online Operation

We now analyze the performance of our method for online gesture detection, including early detection, when applied to unsegmented input streams and trained with the CTC cost function. R3DCNN receives input video streams sequentially as 8-frame clips and outputs class-conditional probabilities after processing each clip. Generally the nucleus of the gesture spans multiple clips, potentially enabling gesture classification before processing all clips.

**Online operation.** Fig. 4 shows ground truth labels and network predictions during continuous online operation on a video sequence collected outside of our previously described dataset. The ground truth in the top row shows the hand-labeled nucleus phase of each gesture. In most cases, both networks—R3DCNN trained with negative log-likelihood ("RNN") and CTC ("CTC"), respectively—predict the correct class before the gesture ends. However, the network trained with CTC produces significantly fewer false positives. The two networks also behave differently when the same gesture is performed sequentially, e.g., observe that three instances of the same gesture occur at $13-17$s and $27-31$s. The CTC network yields an individual peak for each repetition, whereas RNN merges them into a single activation.

**Detection.** To detect the presence of any one of the 25 gestures relative to *no gesture*, we compare the highest current class conditional probability output by R3DCNN to a threshold $\tau \in [0,1]$. When the detection threshold is exceeded, a classification label is assigned to the most probable class. We evaluate R3DCNN trained with and without CTC on the test set with hand-annotated gesture nuclei. We compute the area under the curve (AUC) [12] of the receiver operating characteristic (ROC) curve. The ROC plots the true positive detection rate (TPR)—when the network fires during the nucleus of a gesture—versus the false positive rate (FPR)—when the network fires outside of the nucleus—for a range of threshold values. With the depth modality, CTC results in better AUC (0.91) versus without (0.69) due to fewer false positives. With modality fusion the AUC increases to 0.93.

We also compute the normalized time to detect (NTtD) [12] at a detection threshold ($\tau = 0.3$) with a TPR=88% and FPR=15%. The distribution of the NTtD values for vari-

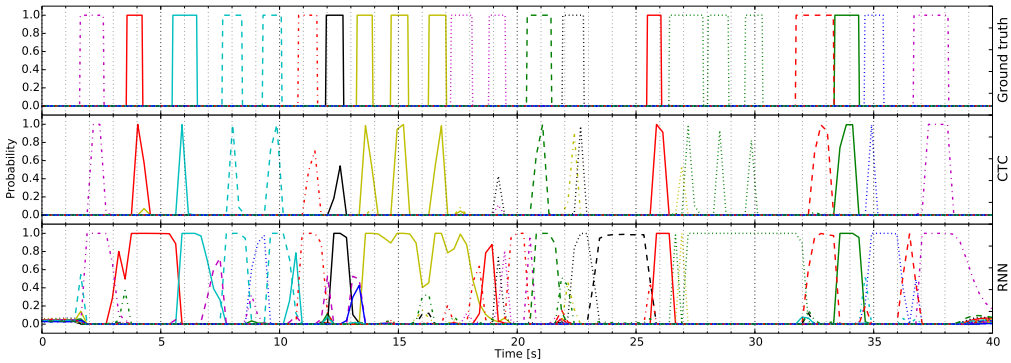


Figure 4: A comparison of the gesture recognition performance of R3DCNN trained with (middle) and without (bottom) CTC. (The *no gesture* class is not shown for CTC.) The various colors and line types indicate different gesture classes.

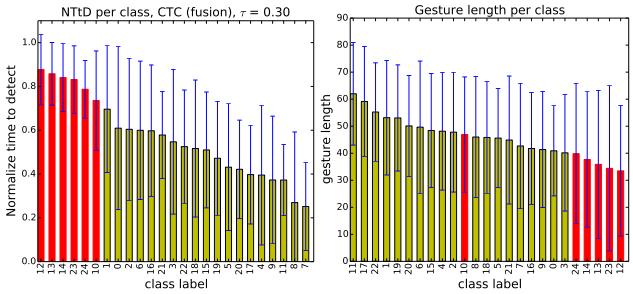


Figure 5: NTtD and gesture length for different classes. Static gestures are marked by red bars.

ous gesture types is shown in Fig. 5. The average NTtD across all classes is 0.56. In general, static gestures require the largest portion of the nucleus to be seen before classification, while dynamic gestures are classified on average within 70% of their completion. Intuitively, the meaning of a static gesture is clear only when the hand is in the final position.

### 5.3. Evaluation on Previously Published Datasets

Finally, we evaluate our method on two benchmark datasets: SKIG [18], and ChaLearn 2014 [5]. SKIG contains 1080 RGBD hand gesture sequences by 6 subjects collected with a Kinect sensor. There are 10 gesture categories, each performed with 3 hand postures, 3 backgrounds, and 2 illumination conditions. Table 5 shows classification accuracies, including the state-of-the-art result established by the MRNN method [26]. Our method outperforms existing methods both with and without the optical flow modality.

The ChaLearn 2014 dataset contains more than 13K RGBD videos of 20 upper-body Italian sign language gestures performed by 20 subjects. A comparison of results is presented in Table 6, including Pigou *et al.* [31] with state-of-the-art classification accuracy of 97.2% and Jaccard score 0.91. On the classification task, our method (with color, depth and optical flow modalities) outperforms this method with an accuracy of 98.2%. For early detection on

Table 5: Results for the SKIG RGBD gesture dataset.

| Method | Modality | Accuracy |
|---|---|---|
| Liu & Shao [18] | color + depth | 88.7% |
| Tung & Ngoc [38] | color + depth | 96.5% |
| Ours | color + depth | **97.7%** |
| MRNN [26] | color + depth + optical flow | 97.8% |
| Ours | color + depth + optical flow | **98.6%** |

Table 6: Results on the Chalearn 2014 dataset. Accuracy is reported on pre-segmented videos. (*The ideal Jaccard score is computed using ground truth localizations, i.e., the class prediction is propagated for the ground truth gesture duration, representing an upper bound on Jaccard score.)

| Method | Modality | Accuracy | Jaccard |
|---|---|---|---|
| Neverova *et al.* [24] | color + depth + skeleton | - | 0.85 |
| Pigou *et al.* [31] | color + depth + skeleton | 97.2% | 0.91 |
| Our, CTC | color | 97.4% | 0.97* |
| Our, CTC | depth | 93.6% | 0.92* |
| Our, CTC | optical flow | 95.0% | 0.94* |
| Our, RNN | color + depth | 96.6% | 0.96* |
| Our, CTC | color + depth | 97.5% | 0.97* |
| Our, RNN | color + depth + optical flow | 97.4% | 0.97* |
| Our, CTC | color + depth + optical flow | **98.2%** | 0.98* |

the color modality, with threshold $\tau = 0.3$ (AUC= 0.98) we observed: TPR=94.8%, FPR=0.8%, and NTtD=0.41, meaning our method is able to classify gestures within 41% of completion, neglecting inference time.

## 6. Conclusion

In this paper, we proposed a novel recurrent 3D convolutional neural network classifier for dynamic gesture recognition. It supports online gesture classification with zero or negative lag, effective modality fusion, and training with weakly segmented videos. These improvements over the state-of-the-art are demonstrated on a new dataset of dynamic hand gestures and other benchmarks.

## References


[1] Bayerische Motoren Werke AG. Gesture control interface in BMW 7 Series. https://www.bmw.com/. 6

[2] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Python for Scientific Computing Conference*, 2010. 4

[3] S. K. Card, G. G. Robertson, and J. D. Mackinlay. The information visualizer, an information workspace. In *ACM CHI*, pages 181–186, 1991. 1

[4] Code Laboratories Inc. Duo3D SDK. https://duo3d.com/. 5

[5] S. Escalera, X. Baró, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon. ChaLearn Looking at People Challenge 2014: dataset and results. In *ECCVW*, 2014. 4, 8

[6] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *Journal of Mach. Learn. Research*, 2008. 3

[7] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scand. Conf. on Im. Anal.*, 2003. 5

[8] D. M. Gavrila. The visual analysis of human movement: a survey. *CVIU*, 73(1):82–98, 1999. 1

[9] A. Graves. *Supervised sequence labelling with recurrent neural networks*. Springer, 2012. 2, 3, 4

[10] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006. 1, 2, 3

[11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*, 2012. 4, 7

[12] M. Hoai and F. De la Torre. Max-margin early event detectors. In *CVPR*, 2012. 2, 7

[13] 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, 3, 6

[14] A. Kendon. Current issues in the study of gesture. In *The biological foundations of gestures: motor and semiotic aspects*, pages 23–47. Lawrence Erlbaum Associates, 1986. 1

[15] K. Kim, D. Lee, and I. Essa. Gaussian process regression flow for analysis of motion trajectories. In *ICCV*, 2011. 2

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

[17] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 5

[18] L. Liu and L. Shao. Learning discriminative representations from RGB-D video data. In *IJCAI*, 2013. 4, 8

[19] G. Marin, F. Dominio, and P. Zanuttigh. Hand gesture recognition with leap motion and kinect devices. In *ICIP*, 2014. 4

[20] R. B. Miller. Response time in man-computer conversational transactions. AFIPS, 1968. 1

[21] S. Mitra and T. Acharya. Gesture recognition: a survey. In *IEEE Systems, Man, and Cybernetics*, 2007. 1

[22] P. Molchanov, S. Gupta, K. Kim, and J. Kautz. Hand gesture recognition with 3D convolutional neural networks. In *CVPRW*, 2015. 1, 2

[23] P. Molchanov, S. Gupta, K. Kim, and K. Pulli. Multi-sensor system for driver's hand-gesture recognition. In *IEEE Automatic Face and Gesture Recognition*, 2015. 1, 2, 6

[24] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. Moddrop: adaptive multi-modal gesture recognition. *arXiv*, 2014. 8

[25] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. Multi-scale deep learning for gesture detection and localization. In *ECCVW*, 2014. 1, 2

[26] N. Nishida and H. Nakayama. Multimodal gesture recognition using multistream recurrent neural network. In *PSIVT*, pages 1–14, 2015. 8

[27] E. Ohn-Bar and M. Trivedi. Hand gesture recognition in real time for automotive interfaces: a multimodal vision-based approach and evaluations. *IEEE ITS*, 15(6):1–10, 2014. 1, 2, 4, 5, 6

[28] M. Osadchy, Y. L. Cun, and M. L. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Mach. Learn. Research*, 8:1197–1215, 2007. 1

[29] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013. 3, 4

[30] F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 2, 5

[31] L. Pigou, A. v. d. Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre. Beyond temporal pooling: recurrence and temporal convolutions for gesture recognition in video. *arXiv*, 2015. 8

[32] M. Ryoo. Human activity prediction: early recognition of ongoing activities from streaming videos. In *ICCV*, 2011. 2

[33] X. Shen, G. Hua, L. Williams, and Y. Wu. Dynamic hand gesture recognition: an examplar based approach from motion divergence fields. *Image and Vis. Computing*, 2012. 2

[34] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition. In *NIPS*, 2014. 1, 2, 5, 6

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale visual recognition. In *ICLR*, 2015. 6

[36] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012. 2

[37] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 1, 2, 3, 4, 5, 6, 7

[38] P. T. Tung and L. Q. Ngoc. Elliptical density shape model for hand gesture recognition. In *ACM SoICT*, 2014. 8

[39] H. Wang, D. Oneata, J. Verbeek, and C. Schmid. A robust and efficient video representation for action recognition. *IJCV*, 2015. 2, 5, 6

[40] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3D action recognition with random occupancy patterns. In *ECCV*, 2012. 2

[41] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. 3

[42] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. In *CVPR*, 2014. 2, 5, 6