Coupled Generative Adversarial Network for Continuous Fine-grained Action Segmentation

Harshala Gammulle Tharindu Fernando Simon Denman Sridha Sridharan Clinton Fookes
Image and Video Research Laboratory, SAIVT, Queensland University of Technology, Australia.
{pranali.gammule, t.warnakulasuriya, s.denman, s.sridharan, c.fookes}@qut.edu.au

Abstract

We propose a novel conditional GAN (cGAN) model for continuous fine-grained human action segmentation, that utilises multi-modal data and learned scene context information. The proposed approach utilises two GANs: termed Action GAN and Auxiliary GAN, where the Action GAN is trained to operate over the current RGB frame while the Auxiliary GAN utilises supplementary information such as depth or optical flow. The goal of both GANs is to generate similar ‘action codes’, a vector representation of the current action. To facilitate this process a context extractor that incorporates data and recent outputs from both modes is used to extract context information to aid recognition. The result is a recurrent GAN architecture which learns a task specific loss function from multiple feature modalities. Extensive evaluations on variants of the proposed model to show the importance of utilising different information streams such as context and auxiliary information in the proposed network; and show that our model is capable of outperforming state-of-the-art methods for three widely used datasets: 50 Salads, MERL Shopping and Georgia Tech Egocentric Activities, comprising both static and dynamic camera settings.

1. Introduction

In this paper, we propose a coupled Generative Adversarial Network approach for continuous action segmentation. Action segmentation is performed as in [23], by predicting the action occurring at every video frame, considering all action classes together with the ‘Background’ (action transitions). We treat the action segmentation process as a generative problem where the generator learns to generate an action code which represents a coded distribution of the action categories present in the current frame.

Most recent works in continuous action segmentation use deep neural networks to process spatio-temporal information, and as such do not require feature engineering or suffer from the inherent deficiencies of hand crafted feature-based methods. Even though deep network based approaches learn to minimise a loss function automatically, manual effort is required to design an effective loss to obtain optimal results [15]. One approach to overcome this is generative adversarial networks [13], which automatically learn a loss function while trying to generate an output indistinguishable from real data. This is achieved using two main components, a ‘generator’ and a ‘discriminator’, where the ultimate goal of the ‘generator’ is to create fake outputs that are difficult for the ‘discriminator’ to distinguish from real ones. Recent research has shown GANs to be effective for
diverse tasks including future frame prediction [29], image synthesis [36], domain adaptation [18, 28], inpainting [51, 34], and visual saliency prediction [33, 10]. However, as yet there are no GAN based methods for detection and recognition problems where the processing of a temporal data stream is required.

We address this by proposing a conditional GAN based approach for continuous action recognition (see system diagram in Figure 1). The proposed method has three main components: the action GAN, the auxiliary GAN and the context extractor. The action GAN takes the current RGB input frame while the auxiliary GAN uses corresponding supplementary information (depth or optical flow images). Both networks are supported by the context information produced by the context extractor, which combines data from both modes to extract additional salient details from the environment to aid recognition. Here, the importance of using environmental context is that it captures information related to the environment and recent actions, which may include details such as items present and interactions between them and the subject.

Both the action and auxiliary GANs aim to generate a similar dense vector representations, an ‘action code’, that represents the current action in the video. The discriminator of each GAN then uses the generated action code and the frame (RGB or auxiliary) to classify whether the action code is generated (‘fake’) or ground truth (‘real’). We reinforce the action code generation process with a classifier loss, forcing generated codes to be informative for the final action classification task that we are interested in.

We aim to capture low-level information in the input frame through the action generator, and high level through the context-extractor through which we also achieve temporal modelling. We treat the input sequence-wise, and synthesised action codes for the previous frame are an input to the context extractor, hence the model is recurrent. In contrast to frameworks which predict start/end frames for actions and operate on images [5] or pre-segmented videos that contain only a single action [44, 11, 12, 16]. Even though such approaches can achieve high accuracy, they represent an oversimplification of the task (as they operate on pre-segmented videos containing one action per video), making them unsuited for real world problems such as detecting threats from a continuous CCTV feed. This limitation is the motivation for researchers to focus on continuous action datasets [38, 45, 47, 20] that contain fine-grain actions.

Numerous action detection and segmentation methods have been introduced over the past few years to address the challenge of continuous action recognition. Some methods are based on extracting hand-crafted features to model fine-grain actions. In [39, 38] Rohrbach et al. introduced methods that utilise pose related and dense trajectory features extracted from HOG, optical flow [22] and motion boundary histograms (MBH) around densely sampled points in the MPII cooking activity dataset [38]. Kuehne et al. [21, 20] introduced temporal convolutional networks (TCN) for fine grain actions in [39, 38] Rohrbach et al. introduced methods based on object centric feature extraction [32, 35]. However these object based methods are limited to actions involving object interactions and are specifically tailored for each dataset, and thus are not easily transferred to a different domain.

Aside from using handcrafted features, approaches have been introduced using deep networks. Singh et al. [45] introduced a multi-stream bi-directional recurrent neural network utilising both spatial and temporal information; while Lea et al. [24] incorporates a spatio-temporal CNN with a constrained segmental model. In [23], the authors have introduced temporal convolutional networks (TCN) for fine grained action detection and segmentation.

However these methods, whether using handcrafted features or deep neural networks, still need manual human effort to perform well; requiring either effort to design feature representations, or to effective loss functions. As a result of this, effort is being channeled to new directions such as Generative Adversarial Networks (GAN). GANs are capa-
imbler of learning outputs that are difficult to discriminate from real examples, and learn a mapping from input to output while learning a loss function to train the mapping. GANs have been applied to solve different computer vision based problems such as future frame [29] or state predictions [55], product photo generation [51], and inpainting [34]. As these methods are image generation methods some studies have sought to add temporal information [52, 17, 41], to extend GANs to classification based approaches [6] and synthesise images from a text description [56]. The works of [52, 17, 41] investigated modelling sequences with a GAN where the generator learns to create a sequence and the discriminator outputs a classification for the entire sequence. However for action segmentation we argue a classification is needed for each frame, rather than for the entire sequence, and hence we cannot directly utilise the above methods.

Several attempts have been made to couple multiple GANs for domain adaptation. Specifically [18] and [28] proposed coupled GAN architectures for generating images in different domains with a joint random vector input. However we are considering multi-modal input streams of the same action representation, and are modelling videos as opposed to a single image. The above stated coupling mechanisms do not utilise multiple input streams, and consider a common random vector input. The coupling is achieved through weight sharing between the 2 generators which is unsuitable in our case as we have diverse inputs (i.e RGB, depth or optical flow) and although they represent the same action, the vastly different modalities necessitate separate generators. Here we propose to perform coupling through a context extractor network, which captures salient information from both modes. Through the context extractor we also obtain a recurrent architecture by using the action code generated at the previous time step as an input in the current time step. Such temporal modelling is essential as we are considering video sequences rather than single images.

3. Methodology

We utilise the conditional GAN [30] (an extension of the GAN), which learns a mapping from the observed image \( x_t \) at time \( t \) and a random noise vector \( z_t \) to \( y_t \) [15]. These GANs have two main components: a Generator \( G \), which creates outputs that it aims to make difficult to distinguish from real data by an adversarially trained discriminator, \( D \), that tries to detect the fake outputs generated by \( G \).

We introduce a conditional GAN based model, coupled action GAN, for continuous fine-grained action segmentation, which couples spatial and temporal information to improve performance. In Section 3.1 we describe the action code format that we train the GAN to create; in Section 3.2 we explain the objectives behind our models and in Section 3.3 we explain the coupled network behaviour.

3.1. Action codes

The generators output an action code representing the action category of each frame. The generator maps dense pixel information to this action code. Hence having a one hot vector is not optimal. Therefore we scale it to a range from 0 to 255, \( y_t \in \mathbb{R}^{1 \times k} \), where \( k \) is the number of action classes in the dataset. Using an integer encoding gives more freedom for the action generator and discriminator to represent each action code as a dense vector representation. The action code can be seen as an intermediate representation of the input frame which is more distinguishable during the classification process, and prior works have shown the value of such representations with GANs [27].

3.2. Objectives

![Figure 2: Visually similar frames from different action classes: background (left) and Inspect_Product (right)](image)

Conditional GANs learn both a mapping from input to output, and a loss to train this mapping. Therefore, they are suitable for problems that require varying loss formulations. The objective for the conditional GAN is defined as,

\[
L_{cGAN}(G, D) = \mathbb{E}[\log D(x_t, y_t)] + \mathbb{E}[\log(1 - D(x_t, G(x_t, z_t)))],
\]

(1)

Many visual recognition tasks critically rely on context [50]. This is especially the case in situations where there are actions that are visually similar (as in Figure 2), but belong to different classes [43, 42, 48, 1]. Therefore, we employ an additional context extractor to enhance the model.

As shown in Figure 1, the inputs for the context extractor, \( \pi \), are the current frames (RGB with the auxiliary data) and the previous action distribution codes obtained by the action generators. Note that the two GANs are formulated and trained in the same manner. The only differences between the two are the input data, and that for the Action GAN (trained with RGB data) a secondary classifier is trained from intermediate outputs to improve learning.

After coupling the generator \( G_{A_t} \) with the context extractor, the adversarial loss for the GANs \( (V_{G_{A_t}, D_t}) \) can be defined as follows,

\[
V_{G_{A_t}, D_t} = \mathbb{E}[\log(D_t(x_t, y_t)) + \mathbb{E}[\log(1 - D_t(x_t, G_{A_t}(x_t, z_t, c_t)))],
\]

(2)
where $c_t$ is the output of $\pi$ at time instance $t$.

Both action generators are trained to output an action code which provides a distribution over all possible action classes. However, our purpose is to classify the frames into discrete action classes. Therefore, we define a classifier that extracts a number of features from intermediate layers of the action generator model ($G_A^1$, i.e. the generator that uses RGB images). Let $\{\theta_0, \theta_1, \ldots, \theta_n\}$ be the features extracted from $n$ layers where $\theta_i = f_i(G_A^1(x_t, c_t))$ is a function that extracts features from the $i^{th}$ layer of the action generator. Then the loss of the multi class classification function $f_c$ is,

$$L_c = \mathbb{E}[\log f_c(f_0(G_A^1(x_t, z_t, c_t)), \ldots, f_n(G_A^1(x_t, z_t, c_t))].$$

(3)

$L_c$ is then used to reinforce the GAN objective with classification error.

$$V_{G_A^1, D_1} = V_{G_A^1, D_1} - \lambda_1 L_c$$

(4)

We use a softmax classifier [3][53] to classify the action class. As all functions are differentiable we train the entire model end-to-end using back-propagation.

### 3.3. Coupling multi-model information

The use of multi-model information benefits a recognition approach as the modes can represent different aspects of the actions [44]. We couple multiple action generators, $G_A^1$ and $G_A^2$, as shown in Figure 1. Here the second GAN is coupled as an auxiliary network, which takes supplementary information. This supplementary information may vary across datasets; for instance we use depth information for the 50 salads dataset [47] and optical flow for MERL shopping [45] and Georgia Tech Egocentric activity [9] datasets. Both GANs aim to generate realistic action codes to fool their respective discriminators using their differing inputs, and the coupled adversarial loss can be defined as,

$$V_{G_A^1, G_A^2, D_1, D_2} = V_{G_A^1, D_1} + V_{G_A^2, D_2}$$

(5)

$$V_{G_A^1, G_A^2, D_1, D_2} = \mathbb{E}[\log(D_1(x_t, y_t))] + \mathbb{E}[\log(1 - D_1(x_t, G_A^1(x_t, z_t, c_t)))] + \mathbb{E}[\log(D_2(x_t', y_t'))] + \mathbb{E}[\log(1 - D_2(x_t', G_A^2(x_t', z_t, c_t)))] - \lambda_1 L_c,$$

(6)

where $x_t'$ is the input to the auxiliary network at time instance $t$, $y_t'$ is the generated action code from $G_A^2$, and $D_1$ and $D_2$ are the respective discriminators for the RGB and auxiliary streams.

$G_A^2$ (auxiliary GAN) differs from $G_A^1$ in that the multi-class classifier (Equation 3) is not applied, and so Equation 2 is used as the adversarial loss for $G_A^2$. As both $G_A^1$ and $G_A^2$ are observing different modalities of the same action at the same time step, and ideally the action codes generated by both $G_A^1$ and $G_A^2$ should be similar. Hence, feeding the classifier with both modalities is redundant. Therefore, we use features from $G_A^1$ only and reinforce only the objective of $G_A^1$ with classification loss.

Network coupling occurs through the loss function. Both GANs try to generate similar codes to represent the current action. The primary stream for segmentation is action GAN and its outputs are used for the final classification result. However both GANs are used as inputs to the context extractor, and so both influence the final decision.

If we try to reinforce the objective of the auxiliary GAN with classification loss the gradients may be smaller (hence ineffective) as $G_A^2$ and the classifier are not directly connected. However through the proposed context extractor we force $G_A^2$ to generate features that are informative for $G_A^1$ for classification. This stabilises the training process (see Fig. 2 in supplementary material showing the convergence). If we use 2 classifiers, one each for $G_A^1$ and $G_A^2$, the objective of $G_A^2$ and the classification objective will attempt to force the representation of $G_A^2$ to be optimal classification performance. Hence rather than allowing $G_A^2$ to learn features complementary to $G_A^1$, this would seek to generate a representation for classifying actions using only the auxiliary stream, discarding it’s relationship with the context extractor and $G_A^1$. Finally, having two classifiers is also redundant as we only seek a single classification output.

### 4. Network Architecture

The input RGB frame is of size of 224 x 224 x 3. We also reshape optical flow and depth maps to sizes 224 x 224 x 2 and 224 x 224 x 1 respectively. The networks used in our model contain modules of the form : 2D convolution, followed by a batch normalisation, and a ReLU activation which we denote convolutional_BatchNorm_ReLu as in [13]. All convolutions are 4 x 4 filters applied with a stride of two such that the output is down sampled by a factor of 2. Specific details for each of the networks (context extractor, action generator, and discriminator) are outlined in Section 4.1, 4.2 and 4.3 respectively. In Section 4.4 we present other details relevant to model training.

#### 4.1. Context Extractor

The context extractor receives two visual inputs: the RGB and auxiliary input frame. Each input is passed through a network with five convolutional_BatchNorm_ReLu layers containing 64, 128, 256, 512 and 512 kernels respectively. The output of the fifth layer (C512) of each stream is flattered and concatenated with the previous action distribution generated by $G_A^1$ and $G_A^2$ for RGB and auxiliary streams respectively. Finally we pass the embedding through a fully connected layer of size 256, generating a context embedding of size 256.
Concatenation is done after encoding the input image as prior concatenation can lead to information loss of the previous action code. When using a single input (see Section 5.4), we omit one of the convolutional_batchNorm_ReLu layers, and use the single available context vector. All other network parameters and layer sizes stay the same.

### 4.2. Action Generators

As the generator network for both $G_{A_1}$ and $G_{A_2}$ we utilise the encoder architecture introduced in [15] as it is effective for visual information embedding. In the network, the input (i.e. RGB frame for $G_{A_1}$ or auxiliary input for $G_{A_2}$), is passed through eight convolutional_batchNorm_ReLu layers. The flattened data is concatenated with the context extractor output before passing through three final fully connected layers. The output of the action generator is of size $k$ units where $k$ is the number of action classes in the dataset. $G_{A_1}$ and $G_{A_2}$ consist of convolutional_batchNorm_ReLu layers containing 64, 128, 256, 512, 512, 512 and 512 kernels followed by fully connected layers, with ReLu activation, of sizes 256, 128 and $k$ where $k$ is the number of action classes in the dataset.

### 4.3. Discriminators

The network either receives the RGB or auxiliary frame as input, depending on which generator it’s paired with, along side the action code. We pass the image input through the two convolutional_batchNorm_ReLu layers before concatenating it with the dense action code representation. The network then outputs whether the input action code is real or fake. A shallow discriminator architecture is used following [40, 31], where the authors found it hard to train deep GANs due to unstable gradients. The discriminator architecture consists of two convolutional_batchNorm_ReLu layers with 64 and 128 kernels followed by a fully connected layer with size 1 and softmax activation.

### 4.4. Network Training

We follow the training method of [15] and alternate between gradient decent passes for the discriminators and the generators, using minibatch SGD (32 examples per minibatch) and the Adam optimiser [19] with an initial learning rate of 0.1 for 25 epochs, and 0.01 for the next 75 epochs. No guidance is provided for the context extractor, and it jointly back propagates with the generators, learning to output informative embeddings. The classifier extracts features from the $8^{th}$, $10^{th}$ and $12^{th}$ layers of $G_{A_1}$, and concatenates them before parsing them to the softmax classifier.

### 5. Evaluation and Discussion

In this Section we present the datasets (Section 5.1) and metrics used (Section 5.2); the performance of the proposed approach compared to state-of-the-art (Section 5.3). Section 5.4 presents an ablation study demonstrating the value of multiple inputs and augmentations such as context.

### 5.1. Datasets

The University of Dundee 50 Salads Dataset [47] contains 50 video sequences of 25 users, each making a salad in two different videos. Each sequence is 5-10 minutes long and obtained from a static RGBD camera pointed at the user. It is a multi-modal dataset including depth and accelerometer data alongside time-synchronised videos. However, we utilise only the video data for evaluation purposes. All seventeen mid-level action classes are used together with the background class.

The MERL Shopping Dataset [45] contains 96 videos (32 subjects, 3 videos per subject), each two minutes long from a static overhead HD camera showing people shopping from grocery-store shelving units. Videos are composed of five action classes other than the background class.

The Georgia Tech Egocentric Activities (GTEA) Dataset [9] is composed of videos recorded from a head mounted camera and includes four subjects performing seven different daily activities. This dataset comprises a dynamic, egocentric camera setting which is significantly different from the static top view of the previous 2 datasets. We utilise 11 action classes defined in [8] including the background class. The evaluation is done as described in [23].

As the supplementary inputs for the auxiliary network, we feed the available depth maps for the 50 Salads Dataset and optical flow images for MERL Shopping and GTEA.

### 5.2. Metrics

To comprehensively evaluate the proposed approach we use segmentation and frame wise accuracy metrics. Frame wise metrics are widely used [24, 23, 47, 20], however as noted by [23] models having similar frame wise accuracies can show large dissimilarities when visualising their performance due to different segmentation behaviour. Therefore, using only frame wise metrics is insufficient to fully describe performance. Considering this, we also use the segmentation metrics: mean average precision with midpoint hit criterion (mAP@mid) [45, 39], Segmental F1 score (F1@k) [23] and segmental edit score (edit) [25].

### 5.3. Results

For all datasets, we consider the Temporal Convolutional Networks (TCN) action detection and segmentation approach of [23] as a baseline. In [23], they introduce two architectures: encoder-decoder TCN (ED-TCN) and dilated TCN; where ED-TCN uses pooling and up sampling to capture long range temporal patterns while dilated TCN uses dilated convolutions.

For the 50 salads dataset we also consider the state-of-the-art methods introduced in [24, 37]. Richard et al. [37] includes statistical length and language modelling to represent temporal and contextual structure, and performs detection and classification jointly. In [24] Lea et al. introduced a fine-grain action segmentation method using spatio-
temporal CNNs able to capture information such as object states, their relationships and their changes over time. We compare to their two proposed models: Spatial CNN and ST-CNN (see Table 1). The TDRN [26] model could be seen as an extension of ED-TCN where the authors replace the temporal convolution layers of the ED-TCN model using deformable temporal convolutions, allowing the model to capture fine-scale temporal details, in contrast to the fixed temporal receptive size of ED-TCN.

For the MERL shopping dataset, we compare the proposed approach to the methods introduced by Singh et al. [45]. We use the results provided by [23], as [23] re-evaluated Singh et al.’s [45] models using segmentation and frame wise metrics. The ‘MSN Det’ results are the sparse set of action detections, while the results for ‘MSN Seg’ are a set of dense (per frame) action segmentations.

For the Georgia Tech egocentric activities dataset, comparison is made to the results provided in [46], who have proposed a CNN network termed ‘Ego ConvNet’ with two streams (a spatial and temporal stream) as introduced in [44]. The remaining systems presented are based on the results obtained through the models Spatial CNN, ST-CNN, Dilated TCN, Bi-LSTM, ED-TCN and TDRN.

When considering the results presented in Table 1 we observe similar frame wise classification accuracies for Spatial CNN [24], Dilated TCN [23], ST-CNN [24], Bi-LSTM [23], ED-TCN [23] TricorNet and TDRN [26]. But significant variations between F1 scores are seen due to over segmentation by the different approaches.

The proposed GAN framework is capable of learning the hierarchical structure of the input frames along with the generated action codes, enabling improved classification of actions. Furthermore, in contrast to the Bi-LSTM, ED-TCN, TricorNet and TDRN models, we model the temporal context as a separate information stream. We believe this enables the proposed model to oversee the evolution of sub-actions and the relationships between them more effectively. This emphasises the importance of capturing auxiliary information available in the dataset and properly localising the present context through the multi-model information streams. These additional data cues along with the ability of GANs to learn a task specific loss allow the proposed model to outperform the state-of-the-arts in both action segmentation and frame wise classification.

Figure 3: Coupled Action GAN prediction results for the 50 Salads dataset.

| Dataset          | Approach     | F1@10,25,50 | edit | mAP@mid | Acc   |
|------------------|--------------|-------------|------|---------|-------|
| S0 Salads        | Spatial CNN  | 32.3, 37.1, 18.9 | 24.8 | NA      | 54.9  |
|                  | IDT+LM       | 44.4, 38.9, 27.8 | 45.8 | NA      | 48.7  |
|                  | Dilated TCN  | 52.2, 47.6, 37.4 | 43.1 | NA      | 59.3  |
|                  | ST-CNN       | 55.9, 49.6, 37.1 | 45.9 | NA      | 59.4  |
|                  | Bi-LSTM      | 52.6, 38.3, 27.0 | 55.6 | NA      | 55.7  |
|                  | ED-TCN       | 68.0, 63.9, 32.6 | 59.8 | NA      | 64.7  |
|                  | TricorNet    | 70.1, 67.2, 33.6 | 62.8 | NA      | 67.5  |
|                  | TDRN         | 72.9, 68.5, 37.2 | 66.0 | NA      | 68.1  |
|                  | Proposed     | 80.1, 78.7, 71.1 | 76.9 | 79.1    | 74.5  |
| MERL Shopping    | MSN Det      | 46.4, 42.6, 25.6 | NA   | 81.9    | 64.6  |
|                  | MSN Seg      | 80.0, 78.3, 65.4 | NA   | 69.8    | 76.3  |
|                  | Dilated TCN  | 79.9, 78.0, 67.5 | NA   | 75.6    | 76.4  |
|                  | ED-TCN       | 86.7, 85.1, 72.9 | NA   | 74.4    | 79.0  |
|                  | Proposed     | 92.8, 91.7, 86.2 | 89.1 | 89.8    | 92.6  |
| GTEA dataset     | EgoNet+TDD   | NA           | NA   | NA      | 64.4  |
|                  | Spatial CNN  | 41.8, 36.0, 25.1 | NA   | 54.1    | 51.4  |
|                  | ST-CNN       | 58.7, 54.4, 41.9 | NA   | 60.6    | 58.3  |
|                  | Dilated TCN  | 58.8, 52.2, 42.2 | NA   | 58.3    | 58.3  |
|                  | Bi-LSTM      | 66.5, 59.0, 43.6 | NA   | 64.0    | 64.0  |
|                  | ED-TCN       | 72.2, 69.3, 56.0 | NA   | 64.8    | 64.8  |
|                  | TricorNet    | 76.0, 71.1, 59.2 | NA   | 74.1    | 70.1  |
|                  | TDRN         | 79.2, 74.4, 82.7 | 74.1 | 70.1    | 78.5  |
|                  | Proposed     | 80.1, 77.9, 69.1 | 72.8 | 78.1    | 78.5  |

Table 1: Action segmentation results for 50 Salads, MERL Shopping and Georgia Tech Egocentric Activities datasets: F1@k is the segmental F1 score, edit is the segmental edit score metric (see [25]), mAP@mid is the mean average precision with mid point hit criterion and accuracy denotes the frame wise accuracy. NA indicates that the metric is unavailable in the respective baseline method.

Figure 4 shows prediction outputs obtained from the coupled action GAN model for the 50 Salads dataset. We observe that there are several areas where the actions have been confused with the background class. For example, the actions such as cut_tomato, cut_cheese, place_cheese_into_bowl. While some classification errors occur, these are typically at the event boundaries where it is difficult to precisely determine action transitions. Overall, we see that all true events are detected and false detected events only last for very short periods of time.

Figure 4: Simplified versions of the coupled action GAN model: unitary GAN (a) and unitary GAN - context (b)

5.4. Ablation Experiments

We evaluate the coupled action GAN model against a series of counter parts which are strategically developed by removing components from the proposed model. The details of these baseline models are as follows.

a. (GA1): A supervised model created by removing the
discriminator and context extractor from the GAN architecture in [1], and back propagating the classification error.

b (G_{A1} + context): The model of (a) with the context extractor added

c (G_{A1} + context + G_{A2}): Couple the model of (b) with G_{A2}. Here we jointly back propagate the classification of the two generator networks.

d (conditional GAN) : unitary GAN model without the context extractor and objective reinforcement. Hence it is the standard conditional GAN model and follows the objective given in Equation [1]

e (unitary GAN - context): unitary GAN model without the context extractor, see Figure [4b]

f (unitary GAN - L_c): unitary GAN model without classification error based objective reinforcement, optimising the objective given in Equation [2]

g (unitary GAN) : coupled action GAN model without the auxiliary network, optimising the objective given in Equation [4] and depicted in Figure [4a]

Using the same experimental settings as in Section 5.3, we present the evaluations for the seven reduced models along with results obtained coupled action GAN in Table 2 for the MERL shopping dataset.

G_{A1} can be considered the simplest model. G_{A1} is not supported by the context extractor, and does not acquire any information regarding the previous time step. Therefore, this naive model simply associates the input frame to an action code without incorporating historic data.

From the results for the model G_{A1} + context, it is evident that the context information has the ability improve performance, with the frame wise accuracy increasing by 16% and the mAP@mid by 17%. With the context extractor the overall model becomes a recurrent model. In the model G_{A1} + context + G_{A2}, the additional information stream improves context information, similar to multimodal streams in MSN Det [45]. However, the model could not achieve the performance of approaches such as MSN Seg [45], Dilated TCN [23] and ED-TCN [23] due to inherent deficiencies [15] when learning with generic loss functions.

We would like to separate the models, G_{A1}, G_{A1} + context and G_{A1} + context + G_{A2} from the rest of the ablation models. The former are generic supervised models which simply map pixels to action classes and learn this mapping through back-propagating an off the shelf classification loss. However with the introduction of the GAN framework with task specific loss function learning, we achieve a significant performance boost compared to these simpler ablation models and the baseline models (see Table [1]). With G_{A1} we perform a softmax classification where we map input pixels to probabilistic labels. This is trivial when the representation and structure of classes are unique, but challenging in continuous action segmentation where background frames appear visually similar to action frames, and action occurrences are related. With the conditional GAN we learn an objective which maps the input to an intermediate representation (i.e action codes) which is easily distinguishable by the classifier. The merit of the intermediate representation is shown by the performance gap between G_{A1} and the conditional GAN. Without sophisticated temporal modelling or very deep feature extraction schemes, even the simplest form of the proposed GAN framework has been able to outperform the baselines by a significant margin. We build upon this observation adding high level context information to the GAN framework and developing a recurrent model to attain sequence modelling.

We observe a gain in performance from conditional GAN to unitary GAN - context due to the use of classification loss based reinforcement of the GAN objective. Furthermore, comparing the accuracies for unitary GAN - context and unitary GAN - L_c, emphasises the importance of proper context localisation in order to recognise complex human actions. The incorporation of context provides rich feature embeddings to drive the action classifier, hence playing a significant role in the proposed framework.

Temporal context localisation and classification loss based reinforcement of the objective further boosts performance of the unitary GAN model. Yet we observe a significant performance improvement from unitary GAN to coupled action GAN, demonstrating the importance of auxiliary input streams for capturing fine grain motion and behavioural patterns. Importantly, we also note that all GAN models in Table 2 outperform the state-of-the-art methods in Table [1] highlighting the benefits that the cGAN architecture offers for continuous fine grained action recognition.

To demonstrate performance if the classifier receives features from both G_{A1} and G_{A2}, we conduct a further experiment as follows. Following the approach for G_{A1} (see Section 3.2), we concatenate G_{A1} and G_{A2} Features as classifier input. Performance for MERL is 90.1 (accuracy) and 87.3 (mAP@mid), both slightly lower than the proposed method. We believe this is due to the added redundant auxiliary features, and the simple softmax classification layer having

| Approach                  | F1@[10,25,50] | mAP@mid | accuracy |
|---------------------------|--------------|---------|----------|
| G_{A1}                    | 49.0, 48.4, 44.2 | 48.3 | 46.1 |
| G_{A1} + context          | 63.6, 62.2, 55.4 | 61.7 | 58.4 |
| conditional GAN           | 86.9, 85.3, 73.7 | 77.1 | 81.3 |
| unitary GAN - context     | 87.2, 85.6, 79.8 | 80.8 | 86.6 |
| unitary GAN - L_c         | 89.9, 86.7, 80.9 | 83.7 | 87.8 |
| unitary GAN               | 89.5, 87.3, 81.0 | 84.3 | 88.2 |
| coupled action GAN        | 92.8, 91.7, 86.2 | 89.8 | 92.6 |

Table 2: Ablation experiment results for MERL Shopping
insufficient capacity to untangle these redundant features. While more capacity could be added, this would increase complexity. Similarly, when classification uses only $G_{A2}$ features; we obtain 89.7 (accuracy) and 86.8 (mAP@mid), indicating the RGB stream is more informative.

### 5.5. Effectiveness of Adversarial + Supervised loss combination

To further demonstrate the discriminative ability of the proposed model we have conduct the following experiment. We select 30 examples from the validation set of the 50 Salads dataset. These examples are chosen from the validation set and include different subjects performing different actions. However, appearance wise all examples exhibit similar characteristics with changes between samples primarily being the hand and object positions. Figure 5a shows the visualisations of the embedding space positions before (in blue) and after (in red) training $G_{A1}$ in the proposed framework. Figure 5b visualises the same embedding space positions for $G_{A1}$ of the ablation model $G_{A1} + context + G_{A2}$, which is trained using a supervised classification loss. Following [4] we extracted activations from layer 5 and applied PCA [49] to plot them in 2D. The respective ground truth class IDs are indicated in brackets.

From Figure 5a it is clear that the frames from the same action class are more tightly grouped by the proposed coupled action GAN, while the supervised learning model is having difficulties learning the common nature of examples within the same action class. This proper grouping leads the introduced model to achieve better classification results. With the GAN learning framework, the generator learns a synthetic objective function that forces it to embed frames from similar action classes closely. This simplifies the task of the action classifier, allowing us to obtain a substantial improvement in performance. The supervised model (Figure 5b) struggles to obtain a proper grouping as is done by the proposed model, where the supervised model embeddings are only loosely grouped after training.

### 5.6. Time Complexity

We evaluate the time consumption of coupled action GAN and the model generates 500 predictions in 24.2 seconds using a single core of an Intel E5-2680 2.50GHz CPU. We also evaluate the unitary GAN ablation model (i.e no auxiliary stream) which makes 500 predictions in 14.1 seconds. We used OpenCV toolbox for optical flow computation and it takes 10.4 seconds for 500 frames. Depth information is already available in the 50 Salads [47] dataset.

### 6. Conclusion

In this paper, we propose a coupled GAN framework for fine grain human action segmentation in video. The proposed model utilises RGB frames and auxiliary information to better model the evolution of human actions within the given video sequence, outperforming state-of-the-art methods on three datasets: the 50 Salads, MERL Shopping and Georgia Tech Egocentric Activities dataset. Evaluations on both static and dynamic cameras with overhead and egocentric view demonstrate the importance of the architectural augmentations proposed in this framework for segmenting fine grain actions. We show the highly beneficial nature of capturing auxiliary information, not only to boost performance but also to ensure the flexibility of the system to adapt to different information cues provided in different datasets. Even though we perform evaluation on unsegmented continuous action datasets, which is comparatively more challenging, the proposed system can be directly applied to pre-segmented datasets for action recognition.

### References

[1] M. Arunothayam, B. Ramachandran, and D. Ponnurangam. *Human Action Recognition and Localization in Video at
Contextual Level, pages 204–207. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[2] M. Aubakirova and M. Bansal. Interpreting neural networks to improve politeness comprehension. Proceedings of EMNLP, 2016.

[3] C. Bishop. Pattern recognition and machine learning (information science and statistics), chapter 3, pages 138–147, 2006.

[4] A. Criminisi, J. Shotton, E. Konukoglu, et al. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Foundations and Trends® in Computer Graphics and Vision, 7(2–3):81–227, 2012.

[5] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. 2010. updated version, available at http://www.di.ens.fr/willow/research/stillactions/.

[6] E. Denton, S. Gross, and R. Fergus. Semi-supervised learning with context-conditional generative adversarial networks. arXiv preprint arXiv:1611.08430, 2016.

[7] L. Ding and C. Xu. Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation. arXiv preprint arXiv:1705.08718, 2017.

[8] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 407–414. IEEE, 2011.

[9] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference On, pages 3281–3288. IEEE, 2011.

[10] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Task specific visual saliency prediction with memory augmented conditional generative adversarial networks. In Applications of Computer Vision (WACV), 2018 IEEE Winter Conference on. IEEE, 2018.

[11] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. Two stream lstm: A deep fusion framework for human action recognition. In Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, pages 177–186. IEEE, 2017.

[12] G. Gkioxari and J. Malik. Finding action tubes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011 IEEE Conference On, pages 1–8. IEEE, 2011.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, pages 448–456, 2015.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

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

[17] A. Kannan and O. Vinyals. Adversarial evaluation of dialogue models. Workshop on Adversarial Training, NIPS 2016., 2017.

[18] M. A. Kiasari, D. S. Moirangthem, and M. Lee. Coupled generative adversarial stacked auto-encoder: Cogasa. Neural Networks, 2018.

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR), 2015.

[20] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.

[21] H. Kuehne, J. Gall, and T. Serre. An end-to-end generative framework for video segmentation and recognition. In Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, pages 1–8. IEEE, 2016.

[22] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008.

[23] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[24] C. Lea, A. Reiter, R. Vidal, and G. D. Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In European Conference on Computer Vision, pages 36–52. Springer, 2016.

[25] C. Lea, R. Vidal, and G. D. Hager. Learning convolutional action primitives for fine-grained action recognition. In Robotics and Automation (ICRA), 2016 IEEE International Conference on, pages 1642–1649. IEEE, 2016.

[26] P. Lei and S. Todorovic. Temporal deformable residual networks for action segmentation in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6742–6751, 2018.

[27] Y. Li, J. Song, and S. Ermon. Infogail: Interpretable imitation learning from visual demonstrations. In NIPS, pages 3815–3825, 2017.

[28] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In Advances in neural information processing systems, pages 469–477, 2016.

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

[30] M. Mirza and S. Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.

[31] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti. Stabilizing gan training with multiple random projections. arXiv preprint arXiv:1705.07831, 2017.

[32] B. Ni, X. Yang, and S. Gao. Progressively parsing interactional objects for fine-grained action detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1020–1028, 2016.

[33] J. Pan, C. Canton, K. McGuinness, N. E. O’Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. arXiv preprint arXiv:1701.01081, 2017.
[34] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[35] K. Ramirez-Amaro, M. Beetz, and G. Cheng. Automatic segmentation and recognition of human activities from observation based on semantic reasoning. In *Intelligent Robots and Systems (IROS 2014)*, 2014 IEEE/RSJ International Conference on, pages 5043–5048. IEEE, 2014.

[36] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[37] A. Richard and J. Gall. Temporal action detection using a statistical language model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3131–3140, 2016.

[38] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, United States, June 2012.

[39] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119(3):346–373, 2016.

[40] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann. Stabilizing training of generative adversarial networks through regularization. *Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[41] M. Saito, E. Matsumoto, and S. Saito. Temporal generative adversarial nets with singular value clipping. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2830–2839, 2017.

[42] M. J. Santofimia, J. Martinez-del Rincon, and J.-C. Nebel. Episodic reasoning for vision-based human action recognition. *The Scientific World Journal*, 2014, 2014.

[43] N. Shapovalova, W. Gong, M. Pedersoli, F. X. Roca, and J. González. On Importance of Interactions and Context in Human Action Recognition, pages 58–66. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[44] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014.

[45] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[46] S. Singh, C. Arora, and C. Jawahar. First person action recognition using deep learned descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2620–2628, 2016.

[47] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013)*, Zurich, Switzerland. ACM, September 2013.