Coarse or Fine? Recognising Action End States without Labels

Davide Moltisanti* Hakan Bilen Laura Sevilla-Lara Frank Keller
University of Bath The University of Edinburgh
dm2460@bath.ac.uk {h.bilen, l.sevilla, frank.keller}@ed.ac.uk

Abstract

We focus on the problem of recognising the end state of an action in an image, which is critical for understanding what action is performed and in which manner. We study this focusing on the task of predicting the coarseness of a cut, i.e., deciding whether an object was cut "coarsely" or "finely". No dataset with these annotated end states is available, so we propose an augmentation method to synthesise training data. We apply this method to cutting actions extracted from an existing action recognition dataset. Our method is object agnostic, i.e., it presupposes the location of the object but not its identity. Starting from less than a hundred images of a whole object, we can generate several thousands images simulating visually diverse cuts of different coarseness. We use our synthetic data to train a model based on UNet and test it on real images showing coarsely/finely cut objects. Results demonstrate that the model successfully recognises the end state of the cutting action despite the domain gap between training and testing, and that the model generalises well to unseen objects.

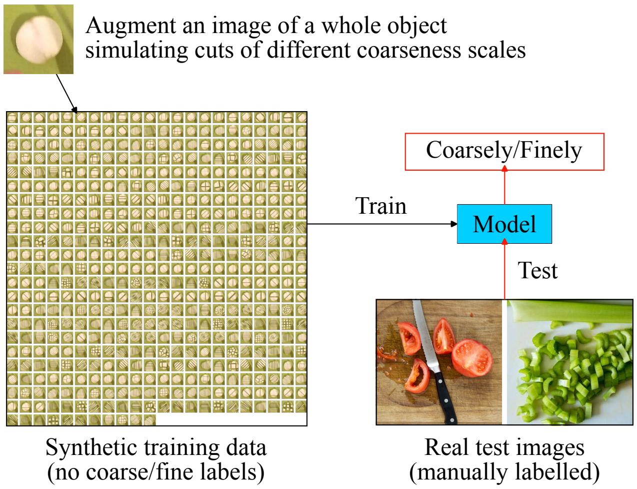


Figure 1. Summary of our work. We aim to recognise the end state of an action, e.g., whether an object is cut coarsely or finely. We assume no labels and propose an object-agnostic image augmentation method to synthesise training data. Our model successfully learns from this synthetic data, as we show by testing on real images and videos, including for unseen objects.

1. Introduction

Action recognition is central to understanding the visual world. When we observe people performing a task (e.g., cooking dinner), we are able to decompose the task in terms of discrete actions (e.g., boiling water, cooking pasta). AI systems that understand videos face the same problem and in response to this, a large literature on action recognition was sprung up [16]. A system recognising actions needs to identify the people and objects involved in the action. Crucially, most actions are characterised by the **state change** of an object. An example is the *cutting* action: independent of which object this action is applied to, it will result in the object being in smaller parts than before. Moreover, the **manner** expressed by an adverb [7, 8, 21] in which an action is performed is crucial to understanding it. For instance, to cut garlic *finely* we would perform a different operation (e.g., we would mince it with a knife) compared to what we would do to cut it *coarsely* (e.g., we might just split it with our hands). Recognising object end states as the result of the way an action is performed is thus important to enable systems to understand more about the action itself. This is a hard endeavour because even for a single action there are a large number of objects that the action can be applied to. Not only do these objects vary visually, the end state will also look differently depending on the type of object and the manner of the action. For example, a finely cut carrot would typically be sliced in small strips, whereas a finely cut clove of garlic would be minced.

In this paper, we present a case study to recognise the end states of one action, cutting, characterised by the main manner in which it can be performed: *coarsely* and *finely*. To focus on the end state recognition problem we do not work on videos and assume only an image depicting the end of the action is given. This allows us to isolate the task

[*] Work done while at the University of Edinburgh

of discerning a coarse cut from a fine cut, without having to worry about video-related issues such as motion, finding the object in a video, etc. We propose a method to generate training data to address the limited availability of datasets labelling coarsely/finely cut objects in images. Using **data augmentation,** we generate a large, high-quality dataset of synthetic images (VOST-AUG) on which we then train our model for end state recognition. This is illustrated in Figure 1. Our augmentation method starts with an image of a whole object and a mask segmenting the object. The approach is object agnostic in that it does not need to know *what* object is in the picture, only *where* it is. We devise several ways to control the simulated coarseness of a cut, which enables us to generate numerous diverse images from a single source. Starting from only 184 images, we generate 90,809 images simulating objects being cut in realistic ways with different coarseness. We also propose a model based on a UNet [27] Encoder-Decoder architecture to take full advantage of our data. Our model is able to achieve 0.856 Mean Average Precision (MAP) on unseen objects, a 4% improvement over the closest baseline.

Our training data is synthetically generated, so it is crucial to test whether the model we propose is able to generalise to real images and videos. Our experiments demonstrate that this is the case, both for an image test set that we collected and annotated (COFICUT) and for the existing video dataset Adverbs in Recipes [21]. In both cases, our model outperforms an existing adverb recognition model [21]. Furthermore, we show that our approach beats a supervised model trained on a portion of COFICUT.

To summarise: (i) We focus on the problem of recognising the end state of an action, which is critical for both action and manner (adverb) recognition. (ii) We propose an object-agnostic augmentation method to synthesise training data for this task. (iii) We present a model based on UNet and train it on our synthetic data. (iv) To evaluate the effectiveness of the synthetic images, we collect a small test set of real images showing coarsely/finely cut objects. Both on this test set and on an existing video dataset, our model achieves good performance, even for unseen objects.

2. Related Work

Adverb Recognition in Videos The closest line of research to action end state recognition is understanding adverbs in videos [7, 8, 12, 21], where models learn to recognise the manner in which actions are performed in a video. In some cases this includes end states, as in "cut coarsely/finely". The approaches of [8, 12, 21] are fully-supervised, while [7] proposes a method that assigns pseudo-labels to the training videos based on the model predictions. However, this still assumes adverb labels are available, since pseudo-labels are assigned from the set of classes in a given dataset. In existing datasets for adverb recognition [7, 8, 21] videos are loosely trimmed and often noisy, without a ground truth localising which frames show the object. This means that learning action end states from such datasets would be difficult. In contrast, we generate training images via augmentation without action, adverb, or object labels. Our model learns to recognise the end state of an action from augmented images in a granular way, i.e., without splitting images into adverb categories. Nevertheless, we show that our model outperforms the adverb recognition model presented in [21], including on videos from the Adverbs in Recipes dataset [21].

Object Attributes in Images Our task also overlaps with the problem of predicting attributes in images [3, 13, 17, 18, 20, 23–25, 28, 31–33], with an important distinction: in object attribute discovery images are typically grouped in a *single* category (e.g., tomato), and the goal is to organise the input group of images into distinct states or attributes (e.g., ripe, raw, peeled, etc). We instead start from an unstructured group of objects and aim to recognise a change in a visual attribute: the coarseness resulting from a cut. In other cases object attribute discovery is addressed from a zero-shot compositional learning perspective, which is a distinct problem compared to action end state recognition. Nevertheless, for completeness we also adapt an attribute discovery model [31] when comparing to state-of-the-art work. We note that the popular MIT-States [13] annotates adjectives including "cut, sliced, peeled, chopped" and especially "thin/thick". However in this dataset "thin/thick" do not necessarily correspond to "coarsely/finely cut", i.e., there are objects such as "sauce, cloud, wall, book, etc" annotated with "thin/thick". For this reason this dataset is not a suitable resource for our problem.

Image Augmentation The success of deep learning on image tasks is in good part due to image augmentation techniques such as cropping, rotation, colour and perspective modifications, etc [22, 29]. Indeed, thanks to these techniques we can expand the visual and semantic diversity of the training data to prevent models from overfitting and enhance their generalisability. In this work we propose a method to augment images, however our method is tailored to synthesise training data from the scratch rather than augmenting an existing training dataset.

3. Probing Existing Methods

In this section we will try to establish how good current retrieval systems are at telling if an object was cut coarsely or finely. We search for food images on Microsoft Bing using the query "{coarsely, finely} cut o", where o is one of 27 objects such as "carrot, garlic, tomato" (see the supplementary material for the full list). We take the top 100 retrieved images, drop duplicates and inspect each image to establish how many images were incorrectly retrieved, i.e., showing a coarse cut when searching for a fine cut, and

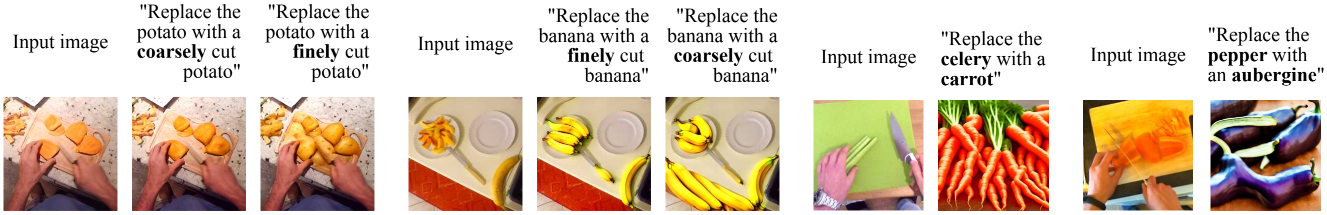

Input image
"Replace the potato with a **coarsely** cut potato"
"Replace the potato with a **finely** cut potato"
Input image
"Replace the banana with a **finely** cut banana"
"Replace the banana with a **coarsely** cut banana"
Input image
"Replace the **celery** with a **carrot**"
Input image
"Replace the **pepper** with an **aubergine**"

Figure 2. Trying to generate images of coarsely/finely cut objects with InstructPix2Pix [1]. Text indicates the prompts used.

vice-versa. Amongst 1,869 images, we found that 42.5% were incorrectly retrieved. This high percentage suggests that retrieval models struggle to distinguish images in these two categories. We do not have internal access to the retrieval system employed by Microsoft Bing, i.e., we do not know for sure whether the search engine uses vision-based text-image retrieval models. If this is the case, then we can ascribe the relatively poor coarse/fine retrieval performance to the fact that text-image retrieval models are typically optimised to distinguish objects into broad classes rather than fine-grained categories. We believe the main reasons for this are the lack of extensive fine-grained labels and the fact that models need to learn beyond the visual appearance of an object, i.e., they need to be able to generalise to recognise coarseness across visually distinct objects.

Generative models are often used in low data regimes to synthesise new images with the desired label. We thus experimented with a generative model to synthesise images of coarsely/finely cut objects. We tested InstructPix2Pix [1], which uses GPT-3 [2] and Stable Diffusion [26] to edit images based on a text prompt. For testing, we used a few images from EPIC Kitchens [4], asking the model to replace a visible object with a finely/coarsely cut version of the same object. Figure 2 shows some examples from this experiment (first two columns). The model generates mostly plausible images, however it just replaces the prompted object with a newly generated version of the same object, ignoring the adverb in the prompt. InstructPix2Pix was trained with hand-made transformation prompts, which means our test instructions are too distinct from the prompts the model was trained on. We further probe this with simpler prompts, asking the model to only replace an object with another one. We still see that the model fails (last two columns) despite the easier prompts. This confirms that the model has not been trained to cover our domain of interest sufficiently. We will show more examples in the supplementary material.

From this Section we conclude that current retrieval methods struggle to differentiate a coarsely cut object from a finely cut one, and that current generative models cannot reliably synthesise images of coarsely/finely cut objects. We therefore propose an image augmentation method that makes it possible to generate synthetic images of coarsely and finely cut objects. We show that this synthetic data can be used successfully to train a classifier that works on real (not synthesised) images of coarsely/finely cut objects.

4. Dataset Creation

4.1. Augmenting to Simulate Cuts

Let us assume we are given an image depicting an object in its whole state and a mask segmenting the object. Our goal is to generate several images depicting the object as if it was cut at different coarseness levels, i.e., from coarsely to finely. We augment the image to achieve this. Specifically, we first remove the object from the image, which we then inpaint to fill the hole left by the removed object using [6]. Next we "break" the object to simulate the result of a cutting action, and overlay the split parts of the object onto the inpainted image to obtain a picture where the object is cut.

Figure 3 illustrates our augmentation method in detail. To break the object, we start sampling n points from the mask. The sampled points act as seeding points to segment object regions, which are obtained by grouping pixels that are closest to one of the n seed points. This is how Voronoi diagrams are built, with the important difference that points are not random but sampled in a way that simulates different human cuts. Specifically, we devise four sampling strategies: *grid:* we sample uniformly both horizontally and vertically, which simulates an object being cut in squares or cubes; *horizontally/vertically:* points are sampled only horizontally or vertically, which simulates objects being cut in vertical or horizontal strips; *diagonally,* where points are sampled along the main or secondary diagonal of the mask, which also simulates objects cut in strips but with an angle (see "Step 1" in Figure 3). Points are evenly spaced initially, however we add random noise to each seeding point to get a more natural looking cut. We next move object regions by a few pixels to "break" the object ("Step 2" in Figure 3), selecting a reference point and pushing each region along the line connecting the region to the reference point. To generate more natural and diverse images, each region is shifted by a number of pixels randomly sampled within an interval. Finally, the moved object regions are overlaid onto the inpainted image without object ("Step 3" in Figure 3).

Figure 4 shows a few synthetic images illustrating how the parameters of our augmentation method affect the re-

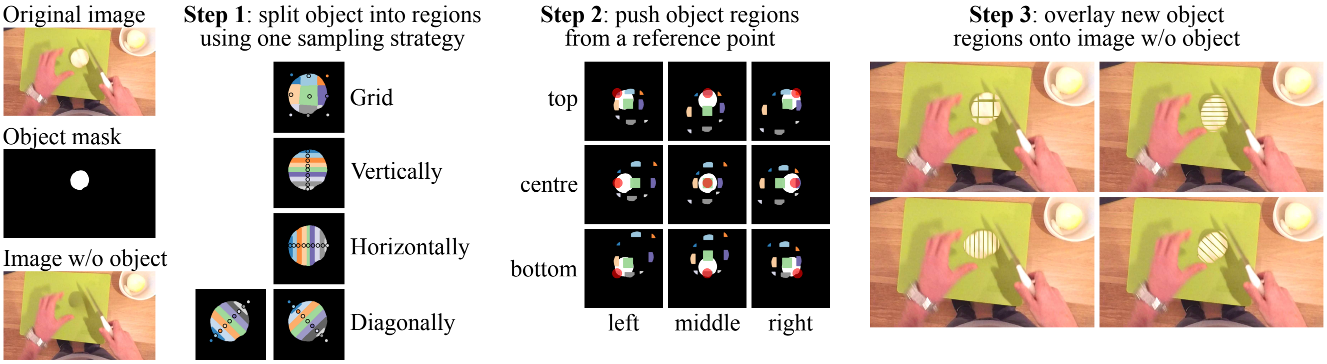
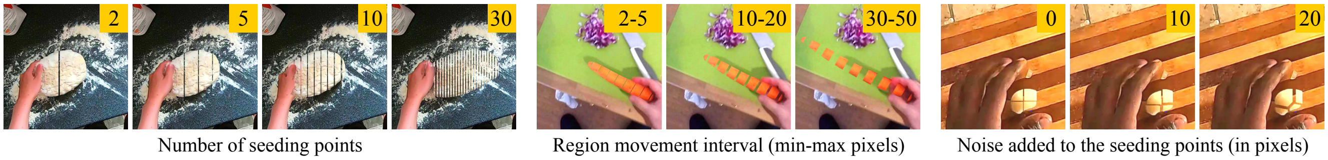

Original image
Object mask
Image w/o object
Step 1: split object into regions using one sampling strategy
Grid
Vertically
Horizontally
Diagonally
Step 2: push object regions from a reference point
top
centre
bottom
left
middle
right
Step 3: overlay new object regions onto image w/o object

Figure 3. Our augmentation method to transform whole objects into cut objects. Given an image and a mask segmenting the object, we first remove the object and inpaint the image to fill the resulting hole (image w/o object, bottom left). We then split the object into regions (**Step 1**). For this we sample n seeding points (nine in this example, indicated by circles) and group object pixels into regions based on their distance to each point, as in a Voronoi diagram. We devise four sampling strategies which affect the topology of the regions and simulate different cut types. We then "break" regions given a reference point (**Step 2**), shown as a red dot, i.e., we push each region away from the reference point along the line connecting the region and the point. Lastly (**Step 3**), we overlay the new regions onto the image w/o object to obtain the final augmented image. We show four examples with reference point (centre, middle) and each of the four sampling strategies.



Number of seeding points

Region movement interval (min-max pixels)

Noise added to the seeding points (in pixels)

Figure 4. Illustrating how the parameters of our augmentation affect the output image. The number of seeding points controls the coarseness of the simulated cut, with fewer/more points corresponding to a coarser/finer cut (left). To obtain more diversified and realistic images we push regions by a random number of pixels sampled within an interval (centre) and add noise to the seeding points (right).

sulting image. The most important parameter is the number of seeding points, which controls the coarseness of the cut (fewer/more points correspond to a coarser/finer cut). The random region movement and the seeding point noise ensure that images are more diverse and natural-looking. Interestingly, these two parameters also affect the perceived *roughness* of a cut, i.e., a greater seeding point noise and a greater region movement will make the cut look rougher or more haphazard. Semantically, the concepts of roughness and coarseness overlap, so it would be difficult and somewhat arbitrary to label the augmented images as coarse or fine based on one or multiple augmentation parameters. For this reason we do not label images into categories. We later show that the difference between an original and augmented image provides a proxy measure to gauge coarseness.

4.2. The VOST-AUG Dataset

We now provide details about how we use augmentation to generate a dataset for this work. We are interested in exploring the potential of a small-scale but high-quality set of annotated images, i.e., we would like to see whether it is possible to train a model to recognise a coarse or fine cut starting from a small set of good images. With this premise, the Video Object Segmentation under Transformations (VOST) dataset [30] is a good resource, as it focuses on actions that significantly transform an object and offers high-quality manual object masks for a few videos from EPIC Kitchens [4] and Ego4D [9]. Furthermore, objects are typically well visible in these datasets thanks to the egocentric viewpoint. VOST annotates 702 videos comprising different actions such as "cut, squeeze, paint", etc. We select only videos labelled with the verb "cut", obtaining 184 videos showing 41 different objects. Video segments are well trimmed in EPIC Kitchens and Ego4D, thus we assume that the first frame in each segment contains the object in its whole state and select the first frame of each segment to build our set of images to augment.[1]

Original images	184	**Objects**	41
Augmented images	90,809	**Objects seen in training**	30
Avg. aug. per image	493	**Objects unseen in training**	11
Training orig. images	96	**Testing orig. images**	84
Training aug. images	47,395	**Testing aug. images**	43,414

Table 1. Summary of the VOST-AUG dataset. Starting from only 184 images we generate 90,809 augmentations showing objects cut at different coarseness scales.

[1] Some segments are part of a repeated action where objects appear partially cut in the first frame. This was not an issue for our augmentations.

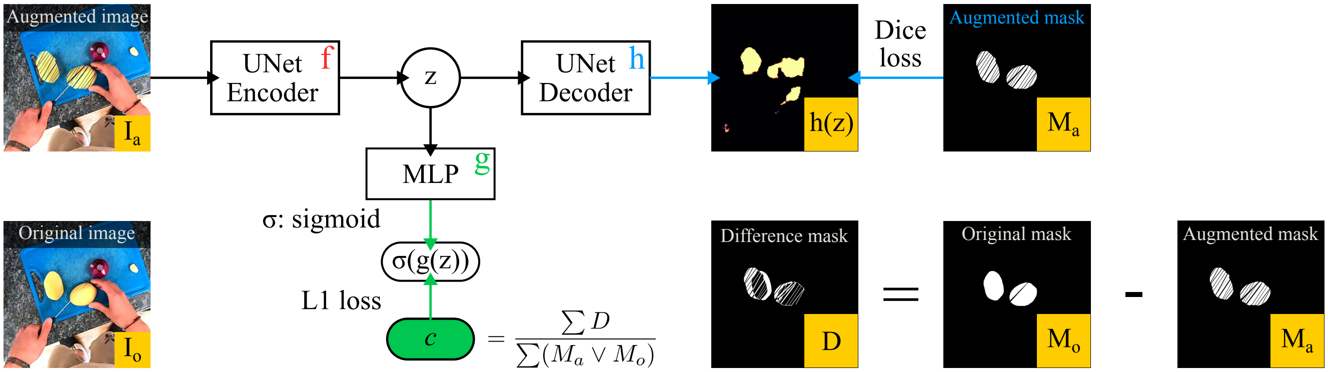


Figure 5. Our model to predict the coarseness of a cut. The model adopts a UNet architecture, where the Encoder bottleneck features z are optimised in two ways. We use an MLP to predict coarseness given z with the L1 loss, using c as target. To learn a stronger z, the UNet decoder adds an auxiliary segmentation task, where we use the augmented object mask as target. The decoder is used only during training. For inference we employ only the Encoder and the MLP output to predict the coarseness of a test image.

As detailed before, there are a few parameters involved in our augmentation method. For this work we augment images taking all combinations of the following parameter values: number of seeding points: 2, 3, 5, 10, 20, 30, 40, 50; seeding points sampling: 'diagonal (main)', 'diagonal (secondary)', 'grid', 'horizontal', 'vertical'; region movement intervals: $[2, 5]$, $[5, 10]$, $[10, 20]$; seeding points noise: 0, 5, 10, 20, 50. For each combination we sample a random reference point among the nine illustrated in Figure 3. With these combinations we generated in total 90,809 augmented images, starting from only 184 original images. On average there are 493 augmented images per original image (some augmentations are rejected if object regions are pushed outside the image bounds, which can happen if the object is near an edge). We split the augmented images in a 70/30 ratio for training/testing, where all augmented images from a given source are either in the train or the test split. We call the set of augmented images the **VOST-AUG** dataset. Table 1 provides a summary of our dataset. We show more examples in the supplementary material.

5. Model

As discussed in Section 4, the coarseness of a cut is mainly controlled by the number of parts a whole object is cut into, however other factors such as the distance between parts and the regularity of their shape also influence the perceived coarseness. We thus design the model based on the difference between an augmented image and its original source: visually, an augmented image will change less/more if the cut is coarser/finer, as we show in Figure 4.

To quantify this, let M_a and M_o be the 2D binary masks segmenting the object in an augmented image I_a and its original source I_o. Let $D = |M_a - M_o|$ be the binary matrix obtained taking the absolute value of the pixel-wise difference between the two masks. We can measure how much an augmented image and its original source differ by comparing their masks. Formally, we have:

$$c(M_a, M_o) = \frac{D}{\sum(M_a \vee M_o)} = \frac{\sum |M_a - M_o|}{\sum(M_a \vee M_o)} \qquad (1)$$

where the denominator normalises the difference between 0 and 1 and ensures that c is independent of the size of the object. Values of c closer to 0/1 indicate a small/large difference between the augmented and the original image, which in turn correspond to a coarser/finer cut.

With the above definition, we can now introduce our model to learn c from I_a to discern the coarseness of a cut. In principle, a model trained with a regression objective such as the L1 loss could be sufficient for this task. However, as we will show in Section 6, it is hard for a model to solve this task without extra guidance due to the subtle differences between the numerous images augmented from a single source. We thus propose an Encoder-Decoder model based on UNet [27]. UNet was designed for medical imaging segmentation, where small-scale details are crucial, thus it is particularly suited for our problem. Our model is depicted in Figure 5. The Encoder f receives in input an augmented image I_a and outputs the bottleneck features $f(I_a) = z$. We optimise z in two ways: firstly, we feed z to an MLP g which outputs a scalar, and use the L1 loss to learn c: $\mathcal{L}_{L1} = |\sigma(g(z)) - c|$, where σ is the sigmoid function[2]. The decoder learns a segmentation mask from the bottleneck features z via skip connections with the Encoder. In our case, the output $h(z)$ is a one-channel image with the same shape as the input. We optimise $h(z)$ with the Dice loss [19], using the augmented object mask as target:

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum\big(M_a \odot h(z)\big) + \epsilon}{\sum\big(M_a + h(z)\big) + \epsilon} \qquad (2)$$

[2] z has shape $(2048, u, v)$, we average along dimensions u and v before feeding it to the MLP.

where ϵ is a small constant for numerical stability. The combined loss to train our model is the sum of the two losses with equal weight, i.e. $\mathcal{L} = \mathcal{L}_{L1} + \mathcal{L}_{Dice}$. The purpose of the Decoder is to provide an auxiliary segmentation task that strengthens the Encoder representation. As is typical with Encoder-Decoder architectures, the Encoder must generate a high-quality representation for the Decoder to effectively solve the dense segmentation task. In other words, the Decoder and the segmentation task provide the "extra guidance" to focus on nuanced differences and learn a better representation for our main task, the coarseness cut estimation. We also use the original images and masks for training. In this case $c = 0$, whereas the target for $\mathcal{L}_{Dice}$ is M_o instead of M_a. We use the Decoder only during training. For inference we only employ the Encoder f and the MLP g and use the output $\sigma(g(f(x)))$ to predict the coarseness of the cut in the image x, where values closer to 0/1 indicate a coarse/fine cut as per c's definition. Our model is able to learn the differences that define the coarseness of a cut by seeing only a single image at the time rather than both the original and the augmented image. This is advantageous as we do not need a reference image for testing. Importantly, our model is object agnostic, so it does not require object labels and does not need an object mask during inference, which makes the model more useful in a real-world setting. We will evaluate the model on real-world (i.e., non-augmented) data in Section 6.1.

6. Experiments

Implementation Details We employ ResNet50 [11] pre-trained on ImageNet [5] as our backbone for all experiments and baselines. Models are trained with the ADAM optimiser [14] for 300 epochs with learning rate $1e-4$, weight decay $5e-5$, batch size 64, dropout 0.1 and no batch normalisation. The MLP in our model has one hidden layer. Input images are resized to 224×224. All experiments are conducted on a single 12GB NVIDIA GeForce RTX 3060.

Evaluation Metric We report Mean Average Precision (MAP) with macro average (the two classes have equal weight). We report MAP globally as well as for seen/unseen objects (except on AIR, where we do not have object labels). For evaluation on VOST-AUG, we group images augmented from the same source and assign them coarse/fine labels based on the median value of c. To clarify, let $\mathcal{I}^A = (I_a^i, i = 1 \dots N)$ be the sequence of N images augmented from the same source I_o, and let $\mathcal{C} = (c(M_a^i, M_o), i = 1 \dots N)$ be the sequence containing the c values obtained from the corresponding augmented masks (see Equation 1). We label each augmented image as follows:

$$y(I_a^i) = \begin{cases} 0 \quad (coarse) & \text{if} \quad c(M_a^i, M_o) \leq \tilde{\mathcal{C}} \\ 1 \quad (fine) & \text{if} \quad c(M_a^i, M_o) > \tilde{\mathcal{C}} \end{cases} \tag{3}$$

where $\tilde{\mathcal{C}}$ denotes the median of $\mathcal{C}$.

Total images	1,869	**Objects**	27
Finely cut images	1,211	**Objects seen in training**	19
Coarsely cut images	658	**Objects unseen in training**	8

Table 2. Summary of the COFICUT evaluation dataset.

Baselines To the best of our knowledge no prior work has focused on our problem with this setting. The closest line of work is adverb recognition in video [7, 12, 21]. We compare against [21], who propose two methods to recognise adverbs termed "CLS" and "REG" . We adapt this model as follows, using the same backbone we use for our model. For CLS, we label training images as coarse/fine as we do for testing on VOST-AUG (see Equation 3). CLS is then a standard classification baseline where the model is trained with binary cross entropy (BCE), i.e., we optimise $\sigma(g(z))$ with the BCE loss (see Figure 5). We also train the CLS model by splitting images based on the number of seeding points instead of the c value, i.e., in Equation 3 we replace $\mathcal{C}$ with $\mathcal{S} = (s_i, i = 1 \dots N)$ and $c(M_a^i, M_o)$ with s_i, where s_i is the number of seeding points used to generate the i-th augmented image. Test images in VOST-AUG are still split as in Equation 3 to compare all models equally.

For REG in [21] verb-adverb video-text embeddings are used to build a regression target. This is sensible when verbs vary, i.e., when there are samples annotated with different verbs for a given adverb. This is not the case in our setting as we only have one verb (cut). We thus adapt REG by using the c values as regression target. This is essentially the same as training our model without the Decoder and the segmentation task, so REG serves also as an ablation study for our full model. All models are trained on VOST-AUG. As we only have two classes, we also provide a random baseline to provide a lower bound. In this case the mean average precision equals the support size of the positive class (testing images with the "fine" label).

We also adapt CANet-CZSL [31], a model for compositional zero-shot attribute learning. We fine-tune the model pre-trained on MIT-States [13], training the model to recognise two attributes: "coarse" and "fine".

6.1. Datasets

COFICUT We collect a set of food images from Microsoft Bing. We start querying "{coarsely, finely} cut o", where o is an object from the list of objects in VOST-AUG, labelling each image with either "coarse" or "fine" based on the query. We take the top 100 retrieved images. We drop duplicates and manually review all images discarding irrelevant results, adjusting their labels to ensure that each image is correctly annotated (as shown in Sec-

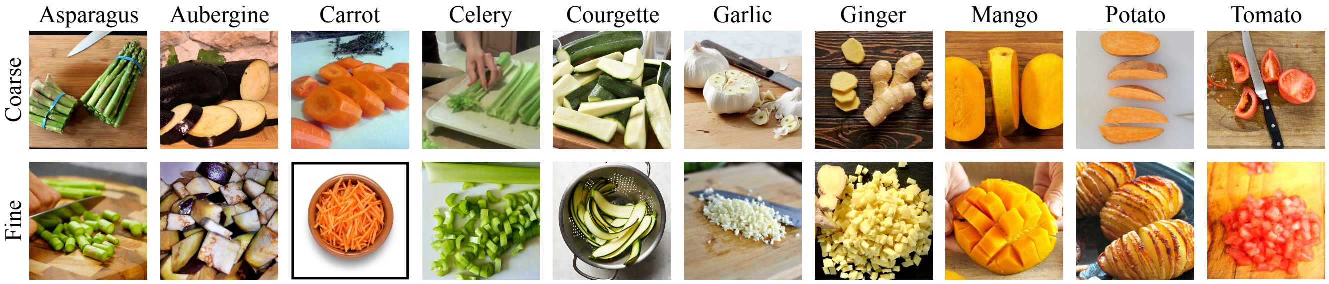


Figure 6. Samples from COFICUT, the dataset of coarsely/finely cut food images we collect for evaluation.

tion 3, the retrieved images were often relevant to the opposite adverb). We remove objects altogether when there was no visible difference between the coarse and fine images. After reviewing, we retain 1,869 images (1,211 labelled as "finely" and 658 labelled as "coarsely") showing 27 different objects (of which 8 are not seen in training). We name this dataset **COFICUT** (COarse-FIne CUT Food Images), which is summarised in Table 2. This dataset is used only for evaluation. Despite its small scale, images have different viewpoints and style than those seen in training, i.e., in training images are all from a first-person point of view (PoV), whereas in testing they are mostly from a third-person PoV. Training images are daily-life captures, whereas COFICUT images are a mix of product pictures, still frames from vlogs or recipe pictures, with very different lighting and style, as illustrated in Figure 6. Above all, training images are synthetic augmentations, whereas test images are real examples of cut objects. For these reasons, we believe COFICUT is a challenging benchmark. COFICUT, VOST-AUG and our code are available at github.com/dmoltisanti/coficut-cvprw24.

Other Datasets We also evaluate models on the test split of the VOST-AUG dataset and the video dataset Adverbs in Recipes (AIR) [21]. As we do not have real binary labels for VOST-AUG, evaluation on VOST-AUG should be seen more as a sanity check rather than a benchmark for comparison. AIR annotates 10 adverbs in instructional videos. We select videos labelled with either "coarsely" or "finely" and one of the following verbs: "chop, cut, mince, grind, grate", for a total of 992 videos. Like in COFICUT, the PoV in AIR is different from that in VOST-AUG. Furthermore, the nature of the videos (instructional) introduces additional diversity, e.g., people explaining their actions are often visible, and videos contain jump cuts and irrelevant content. For this reason, evaluation on AIR is particularly challenging as the models we test are image-based and there is no ground truth localising objects temporally. To test a video in AIR we sample two frames per second and rank the predictions obtained for each frame, aggregating the scores by taking the average of the top 5% scores.

	COFICUT			VOST-AUG			AIR [21]
Model	All	Seen	Unseen	All	Seen	Unseen	All
Random	0.648	0.605	0.759	0.500	0.500	0.500	0.613
CLS [21]	0.692	0.684	0.723	0.625	0.631	0.589	0.623
CLS$_s$ [21]	0.660	0.626	0.742	**0.659**	**0.669**	**0.604**	0.619
REG [21]	0.722	0.702	0.778	0.575	0.584	0.538	0.621
CANet [31]	0.710	0.686	0.811	0.492	0.487	0.535	0.617
Ours	**0.777**	**0.741**	**0.856**	0.561	0.564	0.556	**0.632**

Table 3. Results obtained training models on VOST-AUG. The reported metric is MAP (Mean Average Precision) with macro averaging, where the two classes have equal weight. We report the performance of a random baseline which is equal to the support size of the positive class (the "fine" class). "All/Seen/Unseen" refers to performance evaluated respectively on all images and images showing objects seen/unseen in training.

Model	Training dataset	All	Seen	Unseen
Random	-	0.648 ± 0.000	0.605 ± 0.000	0.759 ± 0.000
BCE	COFICUT	0.447 ± 0.026	0.447 ± 0.026	-
CLS [21]	VOST-AUG	0.693 ± 0.040	0.688 ± 0.045	0.727 ± 0.091
CLS$_s$ [21]	VOST-AUG	0.665 ± 0.031	0.633 ± 0.030	0.751 ± 0.083
REG [21]	VOST-AUG	0.725 ± 0.017	0.706 ± 0.018	0.782 ± 0.053
CANet [31]	VOST-AUG	0.713 ± 0.026	0.690 ± 0.030	0.811 ± 0.061
Ours	VOST-AUG	**0.779** ± 0.005	**0.744** ± 0.025	**0.856** ± 0.039

Table 4. Results obtained with 5-fold cross validation on COFICUT. The reported metric is mean ± std MAP (classes have equal weight). Models trained on VOST-AUG were only tested on the five different folds, while BCE is a classification baseline where a model with the same backbone as the others is trained using the labels available on COFICUT.

6.2. Results

Table 3 compares the performance of the models trained on VOST-AUG and tested on COFICUT, VOST-AUG, and AIR. We note that all models surpass the random baseline on all datasets (except CLS and CANet [31] on some metrics and datasets), which validates our augmentation method: models can tell a coarsely cut object from a finely cut one after being trained on synthetic images without coarse/fine labels. Recall from Table 2 that we generated VOST-AUG's training set from only 96 original images.

We highlight that COFICUT is the most appropriate benchmark for this task as it collects manually reviewed real images of cut objects with a significant visual domain gap. Despite such gap, the diversity of our augmented images allows us to successfully train a model to recognise coarseness in out-of-domain images. In particular, our model achieves the best results by a large margin. This is thanks to the auxiliary task introduced with the UNet decoder, which helps the backbone to focus on the minute details that distinguish the coarseness of a cut. This is evident comparing the regression model (REG) with our model, since REG is essentially an ablation of our model where we discard the Decoder and the segmentation task. Our model is better than REG on the realistic datasets, COFICUT and AIR. This validates the idea of adding the extra task to provide auxiliary guidance. The performance gain for unseen objects further highlights the ability of our model (which is object-agnostic) to generalise well despite the visual gap.

On VOST-AUG we note that CLS (training images split according to c) and CLS_s (split according to seeding points) achieve the best performance. This is not surprising as the model is trained to separate images in the same (CLS) or similar (CLS_s) way as they are split for testing. However, on the remaining datasets both CLS variants rank lowest. Performance on COFICUT unseen objects is even lower than the random baseline, which indicates that the model struggles to generalise. This suggests that splitting images into two classes in our setting is a sub-optimal choice since we have a continuum of simulated cuts ranging from very thin to very coarse, without a neat separation into two classes. We also note that CANet [31] achieves decent results on COFICUT, but performs worse than the random baseline on VOST-AUG on the all/seen metrics. As mentioned before, attribute learning is a different task, so it is difficult for the model to perform well in our distinct setting.

On AIR we observe that performance is poor across all models and closer to the random baseline. This is due to the fact that AIR is a video dataset, so without a ground truth localising the object it is difficult for any model to effectively predict the coarseness of the object shown in the video.

Training on COFICUT We now check whether it would be possible to successfully train a binary classifier on COFICUT. We train the same backbone employed for the other models with binary cross entropy, using the labels available on COFICUT. Given its small size, we conduct this experiment with 5-fold cross validation, comparing against models trained on VOST-AUG by testing them on each fold as well. From Table 4 we see that the model trained on COFICUT ("BCE" in the Table) severely under-performs, with results well below the random baseline (there is no unseen MAP for BCE since all objects are now seen in training). This was expected as COFICUT contains in total 1,869 images, so the model overfits to the training set. However, this also shows that coarseness classification is not a trivial problem and that large training datasets are necessary. Instead of manually annotating images, our augmentation method allows to automatically generate a high-quality, large training dataset that models can successfully learn from. We also note that results with our methods are more robust as they exhibit a lower variation (MAP std is lowest).

7. Conclusion

We addressed the problem of recognising the end state of an action expressed by the manner in which it is performed. We explore this focusing on the cutting action, proposing an approach to detect whether an object is cut *coarsely* or *finely*. We devise an effective image augmentation method to simulate an object being cut at different coarseness levels and in different ways. Starting from only 96 images, we were able to synthesise 47,395 images to train models to successfully recognise whether an object is cut finely or coarsely, without labels. Despite being trained on synthetic images, models achieve good performance on real images and even on unseen objects. We also proposed a model to better leverage the data, boosting performance by over 4%.

Limitations Our augmentation method does not analyse the input scene, and as a result the synthesised image might sometimes look unrealistic. Also, objects may be cut while being held by hand mid-air, in which case the augmentation method produces an image of "levitating" object pieces, as we show in the supplementary material. Scene understanding or affordance approaches [10] could be employed to alleviate this issue. We also need a good object mask to synthesise good images. Recent segmentation models (e.g. Segment Anything [15]) could help lifting this requirement, though objects would still need to be localised (e.g. providing a 2D point or a text prompt describing the object).

Future Directions Being object agnostic, our augmentation method can be adapted to synthesise images where the end state of an object affects its geometry and shape. For example, our method could be extended to predict the *completeness* of a cut, i.e., telling whether an object is *fully* or *partially* cut. Other directions include adapting the augmentation method to synthesise videos and use the augmented data to instil knowledge in retrieval and generative models.

Acknowledgements Research funded by UKRI through the Edinburgh Laboratory for Integrated Artificial Intelligence (ELIAI) and the Turing Advanced Autonomy project.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Neural Information Processing Systems (NeurIPS)*, 2020. 3

[3] Chao-Yeh Chen and Kristen Grauman. Inferring analogous attributes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2022. 3, 4

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6

[6] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[7] Hazel Doughty and Cees G. M. Snoek. How do you do it? fine-grained action understanding with pseudo-adverbs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 6

[8] Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, and Dima Damen. Action modifiers: Learning from adverbs in instructional videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[9] Kristen Grauman et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[10] Mohammed Hassanin, Salman Khan, and Murat Tahtali. Visual affordance and function understanding: A survey. *ACM Computing Surveys (CSUR)*, 2021. 8

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[12] Thomas Hummel, Otniel-Bogdan Mercea, A Koepke, and Zeynep Akata. Video-adverb retrieval with compositional adverb-action embeddings. In *British Machine Vision Conference (BMVC)*, 2023. 2, 6

[13] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 6

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*, 2015. 6

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *International Conference on Computer Vision (ICCV)*, 2023. 8

[16] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *International Journal of Computer Vision (IJCV)*, 2022. 1

[17] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[18] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[19] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Conference on 3D vision (3DV)*. Ieee, 2016. 5

[20] Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[21] Davide Moltisanti, Frank Keller, Hakan Bilen, and Laura Sevilla-Lara. Learning action changes by measuring verb-adverb textual relationships. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 6, 7

[22] Alhassan Mumuni and Fuseini Mumuni. Data augmentation: A comprehensive survey of modern approaches. *Array*, 2022. 2

[23] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[24] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *European Conference on Computer Vision (ECCV)*, 2018.

[25] Zhixiong Nan, Yang Liu, Nanning Zheng, and Song-Chun Zhu. Recognizing unseen attribute-object pair with generative model. In *AAAI Conference on Artificial Intelligence*, 2019. 2

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. 2, 5

[28] Nirat Saini, Khoi Pham, and Abhinav Shrivastava. Disentangling visual embeddings for attributes and objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[29] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 2019. 2

[30] Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the "object" in video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4

[31] Qingsheng Wang, Lingqiao Liu, Chenchen Jing, Hao Chen, Guoqiang Liang, Peng Wang, and Chunhua Shen. Learning conditional attributes for compositional zero-shot learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 6, 7, 8

[32] Xiaoyang Wang and Qiang Ji. A unified probabilistic approach modeling relationships between attributes and objects. In *International Conference on Computer Vision (ICCV)*, 2013.

[33] Yang Wang and Greg Mori. A discriminative latent model of object classes and attributes. In *European Conference on Computer Vision (ECCV)*, 2010. 2