

Method and system for processing a target image

FIELD OF INVENTION

5 **[0001]** The present disclosure relates broadly, but not exclusively, to methods and systems for processing a target image.

BACKGROUND

10 **[0002]** The task of semantic segmentation is to predict pixel-level category labels to given scenes. Although deep neural networks (DNNs) have been widely adopted in the field of semantic segmentation, attaining state-of-the-art performance requires vast quantities of labeled data. It has also been shown that the performance of DNNs is sensitive to the domain misalignment due to the variance in illumination, appearance, viewpoints, back-
15 grounds, etc. between the training and test data.

[0003] One straightforward solution is to annotate more data under the test environment to retrain the segmentation model. However, semantic segmentation is a dense prediction task where it is expensive to manually annotate the pixel-wise labels for data collected in
20 new environments.

[0004] To solve this problem, domain adaptation techniques have been proposed, which aim at transferring the knowledge learned from a source domain (e.g. one or more source images with labelled data forms a source domain from which learned knowledge of each
25 pixel in the one or more source images may be applied to a target domain) to a target domain (e.g. one or more target images with labeled or unlabeled data make up a target domain for training a segmentation model) with minimal performance loss. Domain adaptation may be broadly divided into semi-supervised (some images in the target domain are labeled, some are unlabeled) and unsupervised (all images in the target domain are unlabeled). Specifically, we focus on unsupervised domain adaptation (UDA) for semantic seg-
30 mentation in this work, which is challenging as the images in the target domain are completely unlabeled.

[0005] Existing techniques to solve UDA problems can be roughly divided into two categories: the adversarial training based methods that explicitly align the distributions of the
35

source and target domains and the self-training based methods that recursively refine the pseudo labels on the target data to retrain the network. However, both methods ignore to model the rich semantic relations between pixels across different images in the target domain (e.g. the similarities and differences between pixels of different images in the target domain).

[0006] Self-training based methods focus more on the output level, aiming at generating reliable pseudo labels based on the most confident predictions on the target data. Adversarial training based methods may consider pixel-level features, but the goal is to align the feature distributions between the source and the target domains, rather than modeling the pixel relations within the target domain.

[0007] A need therefore exists to provide methods and systems that seek to overcome or at least minimize the above mentioned challenges.

SUMMARY

[0009] According to a first aspect of the present disclosure, there is provided a method for processing a target image, comprising: generating, by a first network, one or more pseudo labels for a first image, each pseudo label indicating a class of an associated pixel in the first image, the first image being an augmented view of the target image, wherein the first network is a pre-trained domain adaptive segmentation model; and determining, by a second network, one or more labels for a second image based on the one or more pseudo labels, each label indicating a class of an associated pixel in the second image, the second image being an augmented view of the target image and comprising more noise than the first image, wherein the second network is an untrained domain adaptive segmentation model.

[0010] According to a second aspect of the present disclosure, there is provided a system for processing a target image, the system comprising: at least one processor; and at least one memory including computer program code; the at least one memory and the computer program code configured to, with the at least one processor, cause the system at least to: generate, by a first network, one or more pseudo labels for a first image, each pseudo label indicating a class of an associated pixel in the first image, the first image being an augmented view of the target image, wherein the first network is a pre-trained domain adaptive segmentation model; and determine, by a second network, one or more labels for a second image based on the one or more pseudo labels, each label indicating a class of an associated pixel in the second image, the second image being an augmented view of the target image and comprising more noise than the first image, wherein the second network is an untrained domain adaptive segmentation model.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] Embodiments and implementations are provided by way of example only, and will be better understood and readily apparent to one of ordinary skill in the art from the following written description, read in conjunction with the drawings, in which:

[0013] Fig. 1 is a schematic diagram of a computing device. The computing device can be implemented as a system for processing a target image, according to various embodiments of the present disclosure.

[0014] Fig. 2 is a schematic diagram of a system for processing a target image, according to various embodiments.

[0015] Fig. 3 illustrates an example flow diagram of how a network may be trained with a source image according to various embodiments.

[0016] Fig. 4 illustrates an example flow diagram of how a network may be trained via pseudo-label guided learning according to various embodiments.

[0017] Fig. 5 illustrates an example flow diagram of how a network may be trained via pixel-level contrastive learning according to various embodiments.

[0018] Fig. 6 is a block diagram of a computer device suitable for use as a system for processing a target image as shown in Figs. 1 to 5.

[0019] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been depicted to scale. For example, the dimensions of some of the elements in the illustrations, block diagrams or flowcharts may be exaggerated in respect to other elements to help to improve understanding of the present embodiments.

DETAILED DESCRIPTION

[0020] Embodiments will be described, by way of example only, with reference to the drawings. Like reference numerals and characters in the drawings refer to like elements or equivalents.

[0021] Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

[0022] Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as “monitoring”, “utilizing”, “retrieving”, “providing”, “generating”, “quantifying”, “calculating”, “outputting”, “optimising”, “rebuilding”, “storing”, “mapping”, “checking”, “identifying”, “collecting”, “searching”, “conducting”, “cross-checking”, “aggregating”, “determining”, “regenerating”, “updating”, “comparing”, “adjusting”, “compiling”, “performing”, “obtaining”, “predicting” or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

[0023] In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the method described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the scope of the specification.

[0024] Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices

such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a computer effectively results in an apparatus that implements the steps of the preferred method.

DEFINITIONS

[0025] Semantic segmentation refers to a process in which each pixel of an image is labelled with a corresponding class of what is being represented. It is a process that involves predicting a correct label for every pixel in the image, and supports applications such as handwriting recognition, autonomous driving, picture filterings, virtual makeup, virtual clothing try-ons, and many other similar applications. There are various methods and techniques for semantic segmentation. For example, convolutional neural network (CNN) generally consists of few convolutional and pooling layers followed by few fully connected layers at the end. Unet™ builds on top of a fully convolutional network and generally consists of an encoder which down-samples the input image to a feature map and the decoder which up samples the feature map to input image size using learned deconvolution layers. Other methods include deep neural networks (DNNs), DeepLab, Global Convolution Network (GCN) and Kernel-Sharing Atrous Convolution (KSAC).

[0026] A pseudo label is used to indicate a class of a pixel in an image. It is different from ground truth data (e.g. proper and objective data, including class, relating to each pixel in an image to which, for example, class predictions for a pixel of an image can be compared) in that it is typically a prediction of what class a pixel should belong e.g. a pseudo label may indicate an incorrect class for a pixel. Self-training based methods for semantic segmentation recursively refine pseudo labels on target data of a target image to retrain a network, aiming at generating reliable pseudo labels based on the most confident predictions on the target data.

[0027] An embedding or embedding space is generally a low-dimensional space into which vectors such as feature vectors may be translated. Embeddings make it easier to do machine learning on large inputs like sparse vectors representing words or pixels in images. Ideally, an embedding captures some of the semantics of an input by placing semantically similar inputs close together in the embedding space. An embedding can be

learned and reused across various models for semantic segmentations. A network denoted as 'h', for example a DNN, a segmentation model, or other networks, may be represented as a composite of a feature extractor 'f' and a classifier 'g' (e.g. denoted as $h = g \circ f$), wherein an image is mapped into dense pixel embeddings and a feature vector is generated for each pixel of the image. The feature vector is an abstraction of an image pixel that can be used to characterize and numerically quantify the associated image pixel e.g. a list of numbers used to represent an image pixel. The generated feature vectors can be presented spatially in a feature space (e.g. a pixel feature space since each feature vector is representing a pixel of an image).

[0028] Pixel-wise contrastive learning is a branch of deep metric learning in which a network such as a domain adaptive segmentation model is trained by comparing similarities and differences between pixels. For example, given an anchor point, a network distinguishes a similar (or positive) sample from a set of dissimilar (or negative) samples, in a projected embedding space. In the field of computer vision, the contrast is evaluated based on image feature vectors, wherein an augmented version of an anchor image is viewed as a positive, while all the other images in the dataset act as negatives. A goal of pixel-wise contrastive learning is to learn a better-structured pixel embedding space or feature space, wherein pixels belonging to the same class should be close to each other (e.g. intra-class distances are pulled in and reduced among pixels belonging to a same class), while pixels belonging to different classes should be far away from each other (e.g. inter-class distances are pushed further apart among pixels belonging to different classes).

[0029] InfoNCE, where NCE stands for Noise-Contrastive Estimation, is a type of contrastive loss function used for self-supervised learning. Self-supervised learning refers to a category of methods where we learn representations in a self-supervised way (i.e without labels), and may be applied to image learning as well.

[0030] Various studies have been conducted on different approaches for semantic segmentation. In arXiv:2101.10979, categorical prototypes are used to denoise pseudo labels where both the prototypes and the pseudo labels are updated on-the-fly throughout the training. However, this work did not model the semantic relations between pixels in the

target images. In arXiv:2101.11939, pixel-level contrastive learning for semantic segmentation is adopted in a fully supervised manner where all the training images are labeled, but unsupervised techniques are not discussed.

5 **[0031]** In the present disclosure, unsupervised domain adaptation (UDA) for semantic segmentation is utilized. Domain adaptive segmentation is used wherein only the images in a source domain are labeled e.g. having ground truth data. Pixel-level contrastive learning is then performed in an unlabeled target domain based on pseudo labels generated by a teacher network. Pixel-level contrastive learning is adopted to model the semantic
10 relations between pixels, leading to improved segmentation accuracy. This approach is based on self training, and models the semantic relations between the pixels across images in the target domain.

[0032] Fig. 1 illustrates a schematic diagram of a computing device 100. The device 100
15 can be implemented as a system for processing a target image as discussed herein, according to embodiments of the present disclosure.

[0033] The device 100 at least includes a processor 102 and a memory 104. The processor 102 and the memory 104 are interconnected. The memory 104 includes computer
20 program code (not shown in Fig. 1). The memory 104 and the computer program code are configured to, with the processor 102, cause the device 100 to perform the steps for processing a target image as described in the following paragraphs of the present disclosure.

[0034] Fig. 2 depicts a schematic overview of a system 200 for processing a target image,
25 according to various embodiments. The system 200 comprises a first network or teacher network 202 which is a pre-trained model, for example a domain adaptive segmentation model that has been pre-trained to perform semantic segmentation for images, and a second network or student network 204 which may be an untrained model, for example an untrained domain adaptive segmentation model. It may also be possible for the teacher
30 and student network to be of other types of segmentation models known in the art, as long as the teacher network is pre-trained and the student network is untrained. For example, while Resnet101 is utilized as a feature extractor for the teacher and student network in the present examples, other variations such as VGG or Inception may be used instead. In the system 200, the student network 204 is trained to perform more robustly than the
35 teacher network 202 in semantic segmentation of a target image by jointly learning from

labeled source images, target images with pseudo labels generated by the teacher network 202 and target pixel relations, as will be further elaborated in the present disclosure. In the present disclosure, semantic relations between pixels are proposed to be modelled based on supervised contrastive learning. Intuitively, pixel embeddings that belong to the same semantic class should be more similar to each other than those from different classes. One challenge lies in the sampling of the positive pairs (e.g. pixels belonging to the same class) and the negative pairs (e.g. pixels belonging to different classes) to compute the contrastive loss as the images in the target domain are unlabeled.

[0035] As shown in system 200 of Fig. 2, this issue may be addressed by starting with an existing pre-trained domain adaptive segmentation model e.g. teacher network 202 to train a more robust student network 204. For example, the teacher network 202 may be configured to generate one or more pseudo labels 214 for a first image 210, each pseudo label 214 indicating a class of an associated pixel in the first image 210, the first image 210 being an augmented view of a target image 208, wherein the teacher network 202 is a pre-trained domain adaptive segmentation model. Further, one or more labels for a second image 212 may be predicted or determined by the second or student network 204 based on the one or more pseudo labels 214, each label indicating a class of an associated pixel in the second image 212. The second image 212 may be an augmented view of the target image 208 and comprises more noise than the first image 210. The second or student network 204 may be an untrained domain adaptive segmentation model. It will be appreciated that the first image 210 and second image 212 may comprise any form of augmentation from target image 208, as long as second image 212 comprises more noise than first image 210.

[0036] The pseudo labels generated by the teacher network 202 provide a good estimation or prediction of the correct class of the pixels in the target domain e.g. of the target images 208, 210 and 212. However, as the pseudo labels are inevitably noisy, only those pixels with reliable pseudo labels are sampled to compute the contrastive loss. One traditional solution is to select predictions with high confidence level (e.g. with a high chance that the prediction is accurate) by filtering the generated predictions based on a threshold. For example, with a threshold of 0.9, only pixels whose prediction score is higher than 0.9 are selected. However, it is observed in experiments that pixels selected in this way are less effective as they are typically easy examples that can already be correctly classified without contrastive learning. To improve effectiveness, different augmented views of the

same target image 206 are inputted to the teacher and the student networks, respectively. For example, first image 210 may be a weak augmented view of the target image 208 with random crop, and is passed to the teacher network 202 to generate reliable pseudo labels, while a strong augmented view of the target image 206 with more noise e.g. second image 212 is passed to the student network 204 to create hard examples with incorrect predictions. A class-wise memory bank may be utilized to save high-quality pixel features (e.g. pixel feature space 218) from which a large set of negative examples can be sampled for each anchor.

[0037] Given the labeled source dataset (e.g. from a third image or source image 206) and the unlabeled target dataset (e.g. from target images 208, 210 and 212), we aim to train a segmentation model that is expected to predict robustly on the target dataset without accessing its ground truth labels. Typical self-training techniques mostly train a segmentation model on the labeled source domain, which is next utilized to generate the pseudo labels for the images in the unlabeled target domain. Unlike previous work that targets at pseudo-label refinement, we aim to improve existing domain adaptive segmentation models by enforcing the features of intra-class (and inter-class) pixels to be similar (and dissimilar) to each other in the target domain. To achieve this goal, the teacher-student framework as explained above is adopted. The student network 204 is expected to perform more robustly than the teacher network 202 on the target dataset by jointly learning from labeled source images (e.g. via calculation of and minimizing a resultant pixel-wise categorical cross-entropy loss L_{sup}^s), the target images with pseudo labels (e.g. via calculation of and minimizing an overall loss function L_{pseudo}^t with respect to the pseudo labels 214) and the target pixel relations (e.g. via calculation of and minimizing pixel contrast loss L_{contr}^t) with a teacher-student paradigm. An overall loss of the student network 204 may be given as:

$$L = L_{sup}^s + \alpha L_{pseudo}^t + \beta L_{contr}^t$$

wherein:

L = overall loss of student network

L_{sup}^s = pixel-wise categorical cross-entropy loss

L_{pseudo}^t = overall loss function with respect to pseudo labels generated by teacher network

L_{contr}^t = pixel contrast loss

α, β = parameters to adjust weights of different losses in the overall loss function

[0038] Generally, the student network 204 may be represented as 'h', where 'h' is a composite of a feature extractor 220 (represented as 'f') and a classifier 222 (represented as 'g'), denoted as 'h = g o f'. For the student network 204 to learn from the labelled source domain e.g. from the third or source image 206, training the student network 204 with the third image 206 may be based on a standard optimization with the pixel-wise categorical cross-entropy loss as shown below:

$$L_{sup}^s = -\frac{1}{H \cdot W} \sum_{i=1}^{H \cdot W} \sum_{k=1}^K y_s^{(i,k)} \log(p_s^{(i,k)})$$

wherein:

H, W = height and width of input image

10 K = number of classes

$y_s^{(i,k)}$ = corresponding ground truth semantic label of pixel i (e.g. $y_s^{(i,k)} = 1$ if pixel i belongs to a class k , otherwise $y_s^{(i,k)} = 0$)

$p_s^{(i,k)}$ = prediction score of pixel i belonging to class k

15 **[0039]** Fig. 3 illustrates an example flow diagram 300 of how the second network 204 may be trained with the third image 206 according to various embodiments. At step 302, prediction results for the third image 206 are generated by the second network 204, the third image 206 comprising ground truth data 216, wherein the prediction results for the third image 206 are based on the ground truth data 216. At step 304, the one or more labels that are determined by the second network 204 for the second image 212 are further determined based on the prediction results and the ground truth data 216 of the third image 206. At step 306, a pixel-wise categorical cross-entropy loss based on the prediction results and the ground truth data 216 is calculated for the third image 206, wherein determining the one or more labels for the second image 212 further comprises minimising the pixel-wise categorical cross-entropy loss to improve accuracy of the determination. It will be appreciated that the third image may be one of a plurality of images from the source dataset, wherein in each epoch during training of the second network 204, one image of the plurality of images from the source dataset is randomly sampled to compute the pixel-wise categorical cross-entropy loss.

30

[0040] Given an image in the target domain i.e. second image 212, pseudo labels for the second image 212 may be generated by the teacher network 202. The student network

204 may be guided by the pseudo-labels (e.g. pseudo-label guided learning) to self-train and learn from such unlabeled target domain to make predictions of labels for each pixel of the second image 212. A loss function (e.g. such as KL divergence loss function) may be adopted to minimize the distance between the predictions of the student network 204 and the soft pseudo labels 214 generated by the teacher network 202, as shown below:

$$\text{KL}(\hat{p}_t || p_t) = \frac{1}{H \cdot W} \sum_{i=1}^{H \cdot W} \sum_{k=1}^K \hat{p}_t^{(i,k)} \log \frac{\hat{p}_t^{(i,k)}}{p_t^{(i,k)}}$$

wherein:

H, W = height and width of input image

K = number of classes

10 $\hat{p}_t^{(i,k)}$ = soft pseudo label for a pixel i generated by teacher network

$p_t^{(i,k)}$ = prediction for a pixel i generated by student network

[0041] Following the self-training paradigm, a threshold T (eg., $T=0.95$) may be used to select pixels with reliable hard pseudo labels to train the student network 204. The cate-

15 gorical cross-entropy loss based on the hard pseudo labels $\hat{y}_t^{(i,k)}$ generated by the teacher network 202 may be computed as:

$$L_{ce}(\hat{y}_t, p_t) = -\frac{1}{|\xi(x_t)|} \sum_{i \in \xi(x_t)} \sum_{k=1}^K \hat{y}_t^{(i,k)} \log(p_t^{(i,k)})$$

wherein:

K = number of classes

20 $\hat{y}_t^{(i,k)} = \begin{cases} 1 & \text{if } k = \arg \max_{k'} \hat{p}_t^{(i,k')} \\ 0 & \text{otherwise} \end{cases}$

$p_t^{(i,k)}$ = prediction for a pixel i generated by student network

$|\xi(x_t)|$ = indices of the selected pixels

[0042] Subsequently, the overall loss function with respect to the pseudo labels generated by the teacher network 202 may be given as:

$$L_{pseudo}^t = \text{KL}(\hat{p}_t || p_t) + \lambda L_{ce}(\hat{y}_t, p_t)$$

wherein λ is a balancing factor that was empirically set to 1 for the present example.

[0043] Fig. 4 illustrates an example flow diagram 400 of how the student network 204 may be trained via pseudo-label guided learning according to various embodiments. At step 402, one or more pseudo labels for the second image 212 may be generated by the teacher network 202, each pseudo label indicating a class of an associated pixel in the second image 212. At step 404, a loss function may be applied and then the one or more pseudo labels for the second image 212 may be filtered based on a threshold value (e.g. threshold $T=0.95$) to obtain one or more hard pseudo labels for the second image 212, wherein determining the one or more labels for the second image 212 by the student network 204 may be based on the one or more hard pseudo labels for the second image 212. At step 406, a categorical cross-entropy loss may be calculated based on the one or more hard pseudo labels, and an overall loss function for the pseudo labels may be further calculated based on the categorical cross-entropy loss and the loss function, wherein determining the one or more labels for the second image 212 by the student network 204 may further comprise minimising the overall loss function for the pseudo labels to improve accuracy of the determination.

[0044] The system 200 enables the student network 204 to learn from unlabeled target domain via pixel-level contrastive learning. Existing domain adaptive segmentation models mostly cast this task as a pixel-wise classification problem while ignoring the relationship between pixels. Existing techniques proposed to improve semantic segmentation results by applying pixel-wise contrastive learning in a supervised setting to learn a better-structured pixel embedding space. Intuitively, in a well-structured embedding space, pixels belonging to the same class should be close to each other, while pixels belonging to difference classes should be far away from each other. However, it is challenging to learn such a well-structured embedding space in an unsupervised domain adaptation setting as we do not have access to the ground-truth labels of the pixels in the target domain.

[0045] One solution for the above issue is to use the same threshold T (e.g. threshold $T=0.95$) to select reliable hard pseudo labels from the pseudo labels generated by the teacher network 202, which can be regarded as the ground-truth labels to generate the positive and negative pairs for supervised contrastive learning. However, the pixels selected in this way are likely to be easy examples, as most of them can already be robustly classified by the segmentation model. Therefore, the key to improve the segmentation accuracy is to focus more on the pixels with incorrect predictions. By sampling such hard examples as anchors, we will be able to learn segmentation-aware pixel embeddings that

can be more easily classified by the segmentation model after applying pixel-level contrastive learning.

[0046] To sample hard examples as anchors in the unlabeled target domain, the pseudo labels generated under weak augmentation (e.g. pseudo labels generated by the teacher network 202 for the first image 210) may be utilized to guide the contrastive learning for strong augmented views (e.g. second image 212). More specifically, the weak augmented views of the images in the target domain are inputted to the teacher network 202, and the strong augmented views of the same images are inputted to the student network 204. As strong augmentation introduces more noise, the prediction of the teacher network 202 and the prediction of the student network 204 are difficult to be always aligned with each other. Based on the teacher network 202's output, we select pixels with reliable hard pseudo labels (filtered by threshold T) for anchor sampling. A candidate pixel is considered to be a hard example if the prediction of the student network 204 is different from the pseudo label generated by the teacher network 202. Otherwise, it is considered to be an easy example where the predictions of the student and the teacher networks are the same.

[0047] As previously explained, the student network 204 may be represented as 'h', where 'h' is a composite of a feature extractor 220 (represented as 'f') and a classifier 222 (represented as 'g'), denoted as 'h = g o f'. The feature extractor 220 'f' maps an input image 'x' into dense pixel embeddings f(x), while the classifier 222 'g' is applied on top of 'f' to predict final probability scores denoted as 'h(x)=g(f(x))'. Accordingly, projection head 'g_proj' may be introduced, which is applied on top of 'f' to generate a 256-d l_2 -normalized feature vector v for each pixel such that $v \in V$ where $V = g_{proj}(f(x))$. Thereafter, InfoNCE is adopted to compute the pixel contrast loss as:

$$L_{contr}^t = -\frac{1}{|\mathcal{P}|} \sum_{v \in \mathcal{P}} \log \frac{\exp(v \cdot v^+ / \tau)}{\exp(v \cdot v^+ / \tau) + \sum_{v^- \in \mathcal{N}_v} \exp(v \cdot v^- / \tau)}$$

wherein:

$\mathcal{P}$ = normalized feature vector collection of all anchors selected in an image

$\mathcal{N}_v$ = feature vector collection of all negative examples with respect to v

v = 256-d normalized feature vector for a single anchor

v^+ and v^- = feature vectors for positive and negative examples with respect to v , respectively

$\tau > 0$ is a temperature hyper-parameter

[0048] Fig. 5 illustrates an example flow diagram 500 of how the student network 204 may be trained via pixel-level contrastive learning according to various embodiments. At step 502, one or more pseudo labels 214 for the first image 210 may be generated by the first network 202, each pseudo label 214 indicating a class of an associated pixel in the first image 210, the first image 210 being an augmented view of the target image 208, wherein the first network 202 is a pre-trained domain adaptive segmentation model. At step 504, one or more labels for a second image 212 may be determined by the second network 204 based on the one or more pseudo labels 214, each label indicating a class of an associated pixel in the second image 212, the second image 212 being an augmented view of the target image 208 and comprising more noise than the first image 210, wherein the second network 204 is an untrained domain adaptive segmentation model. At step 506, the one or more pseudo labels 214 may be filtered based on the threshold value (e.g. threshold T) to obtain one or more hard pseudo labels, wherein determining the one or more labels for the second image 212 is based on the one or more hard pseudo labels. At step 508, each pixel of the second image 212 may be compared with one another based on the one or more hard pseudo labels of the first image 210. At step 510, one or more positive and negative pairs may be determined based on the comparison for supervised contrastive learning, a positive pair comprising two pixels of the second image that belong to a same class and a negative pair comprising two pixels of the second image that belong to a different class from each other, wherein determining the one or more labels for the second image 212 is further based on the one or more positive and negative pairs. At step 512, the second network 204 may generate a feature vector for each pixel in the second image 212 based on the one or more pseudo labels. At step 514, a pixel contrast loss may be calculated based on the feature vector for each pixel in the second image 212, wherein determining the one or more labels for the second image 212 further comprises minimising the pixel contrast loss to improve accuracy of the determination.

[0049] Accordingly, the key contributions of the present disclosure are as follows: a domain adaptive segmentation network e.g. second or student network 204 is advantageously enhanced by modeling the semantic relations among the pixels cross images in the unlabeled target domain, and target pixel relations are modelled by supervised contrastive learning. To create hard examples with reliable pseudo labels, the resultant pixel feature space is optimized based on strong image augmentation guided by the pseudo labels generated on the weak augmented views of the same images.

[0050] Extensive experiments are performed utilizing the above-described techniques for adapting from Grand theft Auto 5 (GTA5™) and SYNTHIA™ datasets to the Cityscapes™ dataset. Experimental results show that the proposed methods presented in the present disclosure outperforms the state-of-the-art domain adaptive segmentation methods by a significant margin.

[0051] Fig. 6 is a block diagram of a computer device suitable for use as a system for processing a target image as shown in Figs. 1 to 5. The following description of the computer system / computing device 600 is provided by way of example only and is not intended to be limiting.

[0052] As shown in Fig. 600, the example computing device 600 includes a processor 604 for executing software routines. Although a single processor is shown for the sake of clarity, the computing device 600 may also include a multi-processor system. The processor 604 is connected to a communication infrastructure 606 for communication with other components of the computing device 600. The communication infrastructure 606 may include, for example, a communications bus, cross-bar, or network.

[0053] The computing device 600 further includes a main memory 608, such as a random access memory (RAM), and a secondary memory 610. The secondary memory 610 may include, for example, a hard disk drive 612 and/or a removable storage drive 614, which may include a magnetic tape drive, an optical disk drive, or the like. The removable storage drive 614 reads from and/or writes to a removable storage unit 618 in a well-known manner. The removable storage unit 618 may include a magnetic tape, optical disk, or the like, which is read by and written to by removable storage drive 614. As will be appreciated by persons skilled in the relevant art(s), the removable storage unit 618 includes a computer readable storage medium having stored therein computer executable program code instructions and/or data.

[0054] In an alternative implementation, the secondary memory 610 may additionally or alternatively include other similar means for allowing computer programs or other instructions to be loaded into the computing device 600. Such means can include, for example, a removable storage unit 622 and an interface 620. Examples of a removable storage unit 622 and interface 620 include a removable memory chip (such as an EPROM or PROM)

and associated socket, and other removable storage units 622 and interfaces 620 which allow software and data to be transferred from the removable storage unit 622 to the computer system 600.

- 5 **[0055]** The computing device 600 also includes at least one communication interface 624. The communication interface 624 allows software and data to be transferred between computing device 600 and external devices via a communication path 626. In various embodiments, the communication interface 624 permits data to be transferred between the computing device 600 and a data communication network, such as a public data or private
- 10 data communication network. The communication interface 624 may be used to exchange data between different computing devices 600 which such computing devices 600 form part an interconnected computer network. Examples of a communication interface 624 can include a modem, a network interface (such as an Ethernet card), a communication port, an antenna with associated circuitry and the like. The communication interface 624
- 15 may be wired or may be wireless. Software and data transferred via the communication interface 624 are in the form of signals which can be electronic, electromagnetic, optical or other signals capable of being received by communication interface 624. These signals are provided to the communication interface via the communication path 626.
- 20 **[0056]** Optionally, the computing device 600 further includes a display interface 602 which performs operations for rendering images to an associated display 630 and an audio interface 632 for performing operations for playing audio content via associated speaker(s) 634.
- 25 **[0057]** As used herein, the term "computer program product" may refer, in part, to removable storage unit 618, removable storage unit 622, a hard disk installed in hard disk drive 612, or a carrier wave carrying software over communication path 626 (wireless link or cable) to communication interface 624. Computer readable storage media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to
- 30 the computing device 600 for execution and/or processing. Examples of such storage media include floppy disks, magnetic tape, CD-ROM, DVD, Blu-ray™ Disc, a hard disk drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computing device 600. Examples of transitory or non-tangible

computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computing device 600 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

[0058] The computer programs (also called computer program code) are stored in main memory 608 and/or secondary memory 610. Computer programs can also be received via the communication interface 624. Such computer programs, when executed, enable the computing device 600 to perform one or more features of embodiments discussed herein. In various embodiments, the computer programs, when executed, enable the processor 604 to perform features of the above-described embodiments. Accordingly, such computer programs represent controllers of the computer system 600.

[0059] Software may be stored in a computer program product and loaded into the computing device 600 using the removable storage drive 614, the hard disk drive 612, or the interface 620. Alternatively, the computer program product may be downloaded to the computer system 600 over the communications path 626. The software, when executed by the processor 604, causes the computing device 600 to perform functions of embodiments described herein.

[0060] It is to be understood that the embodiment of Fig. 6 is presented merely by way of example. Therefore, in some embodiments one or more features of the computing device 600 may be omitted. Also, in some embodiments, one or more features of the computing device 600 may be combined together. Additionally, in some embodiments, one or more features of the computing device 600 may be split into one or more component parts.

[0061] It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present disclosure as shown in the specific embodiments without departing from the scope of the specification as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

CLAIMS

What is claimed is:

1. A method for processing a target image, comprising:

generating, by a first network, one or more pseudo labels for a first image, each pseudo
5 label indicating a class of an associated pixel in the first image, the first image being an
augmented view of the target image, wherein the first network is a pre-trained domain
adaptive segmentation model; and

determining, by a second network, one or more labels for a second image based on the
10 one or more pseudo labels, each label indicating a class of an associated pixel in the
second image, the second image being an augmented view of the target image and com-
prising more noise than the first image, wherein the second network is an untrained do-
main adaptive segmentation model.

2. The method of claim 1, further comprising filtering the one or more pseudo labels
15 based on a threshold value to obtain one or more hard pseudo labels, wherein determining
the one or more labels for the second image is based on the one or more hard pseudo
labels.

3. The method of claim 2, further comprising:

20 comparing each pixel of the second image with one another based on the one or more
hard pseudo labels of the first image, and

determining one or more positive and negative pairs based on the comparison for super-
vised contrastive learning, a positive pair comprising two pixels of the second image that
25 belong to a same class and a negative pair comprising two pixels of the second image
that belong to a different class from each other,

wherein determining the one or more labels for the second image is further based on the
one or more positive and negative pairs.

4. The method of claim 3, further comprising generating, by the second network, a
30 feature vector for each pixel in the second image based on the one or more pseudo labels,
and calculating a pixel contrast loss based on the feature vector for each pixel in the sec-
ond image, wherein determining the one or more labels for the second image further com-
prises minimising the pixel contrast loss to improve accuracy of the determination.

35 5. The method of claim 2, further comprising:

generating, by the first network, one or more pseudo labels for the second image, each pseudo label indicating a class of an associated pixel in the second image;

5 applying a loss function and then filtering the one or more pseudo labels for the second image based on the threshold value to obtain one or more hard pseudo labels for the second image, wherein determining the one or more labels for the second image is based on the one or more hard pseudo labels for the second image.

6. The method of claim 5, further comprising calculating a categorical cross-entropy loss based on the one or more hard pseudo labels, and an overall loss function for the
10 pseudo labels based on the categorical cross-entropy loss and the loss function, wherein determining the one or more labels for the second image further comprises minimising a value of the overall loss function for the pseudo labels to improve accuracy of the determination.

15 7. The method of claim 1, further comprising:

generating, by the second network, prediction results for a third image, the third image comprising ground truth data, wherein the prediction results for the third image are based on the ground truth data, and wherein determining the one or more labels for the second image is further based on the prediction results and the ground truth data of the third
20 image.

8. The method of claim 7, further comprising calculating a pixel-wise categorical cross-entropy loss based on the prediction results and the ground truth data for the third image, wherein determining the one or more labels for the second image further comprises
25 minimising a value of the pixel-wise categorical cross-entropy loss to improve accuracy of the determination.

9. A system for processing a target image, the system comprising:

at least one processor; and

30 at least one memory including computer program code;

the at least one memory and the computer program code configured to, with the at least one processor, cause the system at least to:

35 generate, by a first network, one or more pseudo labels for a first image, each pseudo label indicating a class of an associated pixel in the first image, the first image being an augmented view of the target image, wherein the first network is a pre-trained domain adaptive segmentation model; and

determine, by a second network, one or more labels for a second image based on the one or more pseudo labels, each label indicating a class of an associated pixel in the second image, the second image being an augmented view of the target image and comprising more noise than the first image, wherein the second network is an untrained domain adaptive segmentation model.

10. The system of claim 9, wherein the system is further configured to filter the one or more pseudo labels based on a threshold value to obtain one or more hard pseudo labels, wherein the one or more labels for the second image is determined based on the one or more hard pseudo labels.

11. The system of claim 10, wherein the system is further configured to:

compare each pixel of the second image with one another based on the one or more hard pseudo labels of the first image, and

determine one or more positive and negative pairs based on the comparison for supervised contrastive learning, a positive pair comprising two pixels of the second image that belong to a same class and a negative pair comprising two pixels of the second image that belong to a different class from each other,

wherein the one or more labels for the second image is further determined based on the one or more positive and negative pairs.

12. The system of claim 11, wherein the system is further configured to generate, by the second network, a feature vector for each pixel in the second image based on the one or more pseudo labels, and calculate a pixel contrast loss based on the feature vector for each pixel in the second image, wherein determining the one or more labels for the second image further comprises minimising the pixel contrast loss to improve accuracy of the determination.

13. The system of claim 10, wherein the system is further configured to:

generate, by the first network, one or more pseudo labels for the second image, each pseudo label indicating a class of an associated pixel in the second image;

apply a loss function and then filter the one or more pseudo labels for the second image based on the threshold value to obtain one or more hard pseudo labels for the second image, wherein determining the one or more labels for the second image is further based on the one or more hard pseudo labels for the second image.

14. The system of claim 13, wherein the system is further configured to calculate a categorical cross-entropy loss based on the one or more hard pseudo labels, and an overall loss function for the pseudo labels based on the categorical cross-entropy loss and the loss function, wherein determining the one or more labels for the second image further
5 comprises minimising a value of the overall loss function for the pseudo labels to improve accuracy of the determination.

15. The system of claim 9, wherein the system is further configured to:

10 generate, by the second network, prediction results for a third image, the third image comprising ground truth data, wherein the prediction results for the third image are based on the ground truth data, and wherein determining the one or more labels for the second image is further based on the prediction results and the ground truth data of the third image.

15 16. The system of claim 15, wherein the system is further configured to calculate a pixel-wise categorical cross-entropy loss based on the prediction results and the ground truth data for the third image, wherein determining the one or more labels for the second image further comprises minimising a value of the pixel-wise categorical cross-entropy loss to improve accuracy of the determination.
20