EMPLACE: Self-Supervised Urban Scene Change Detection

Tim Alpherts, Sennay Ghebreab, Nanne van Noord

University of Amsterdam
t.o.l.alpherts@uva.nl, s.ghebreab@uva.nl, n.j.e.vannoord@uva.nl

Abstract

Urban change is a constant process that influences the perception of neighbourhoods and the lives of the people within them. The field of Urban Scene Change Detection (USCD) aims to capture changes in street scenes using computer vision and can help raise awareness of changes that make it possible to better understand the city and its residents. Traditionally, the field of USCD has used supervised methods with small scale datasets. This constrains methods when applied to new cities, as it requires labour-intensive labeling processes and forces a priori definitions of relevant change. In this paper we introduce AC-1M the largest USCD dataset by far of over 1.1M images, together with EMPLACE, a self-supervising method to train a Vision Transformer using our adaptive triplet loss. We show EMPLACE outperforms SOTA methods both as a pre-training method for linear fine-tuning as well as a zero-shot setting. Lastly, in a case study of Amsterdam, we show that we are able to detect both small and large changes throughout the city and that changes uncovered by EMPLACE, depending on size, correlate with housing prices - which in turn is indicative of inequity.

Figure 1: Examples of various urban changes in Amsterdam uncovered through EMPLACE.

Introduction

Visual Urban Analytics (VUA) approaches have over the last decade shown potential to identify socio-economic inequity by combining computer vision techniques with street view imagery (Suel et al. 2019; Naik et al. 2017). It has been shown that visual appeal affects citizen well-being through aspects such as greenery (Li et al. 2015), perceived safety (Naik et al. 2014; Ordonez and Berg 2014), or liveliness (Dubey et al. 2016), as well as more direct definitions of liveability (Joglekar et al. 2020; Muller et al. 2022; Batty 2019). Proposed approaches predict either objective socio-economic metrics (Suel et al. 2019, 2021; Law, Paige, and Russell 2019) or subjective perceived attributes (Naik et al. 2014; Dubey et al. 2016) and show potential to support municipalities to keep a grasp on the state of their neighbourhoods (Alpherts, van Noord, and Ghebreab 2023). However all these approaches are static as they use a single temporal datapoint, i.e., a single image per location. As a consequence, a concept that is underexplored in VUA is the notion of urban change. Cities are constantly changing; expanding by adding new neighbourhoods and buildings, whilst existing environments decay and get renovated. Many of these changes require permits or are initiated by the municipality, which leads to awareness of these changes. However, there are also many changes to the urban environment that the municipality is not aware off - changes which can be predictive of the condition of the urban environment.

In fact, recent works within VUA have shown the relationship between urban change and socio-demographic data, by investigating how socio-economic indicators predict neighbourhood improvement (Naik et al. 2017) or proposing methods to use computer vision to detect building construction (Huang et al. 2024). The latter of these two approaches falls within the field of Urban Scene Change Detection (USCD), a subfield of computer vision that revolves around finding changes in sets of street view images. To further investigate the relationship between urban change and socio-demographic data we aim to extend VUA using USCD methods to explore visual changes in neighbourhoods. Unfortunately, current USCD methods are not well-equipped for this. USCD has mostly been used for specific domains such as detecting tsunami damage (Sakurada and Okatani 2015) or autonomous driving (Alcantarilla et al. 2016), lim-

iting their broader use within cities. Moreover, the existing datasets for USCD are relatively small, and use a variety of labelling techniques such as pixel-wise annotations (Sakurada and Okatani 2015; Alcantarilla et al. 2016), further limiting direct application or adaptation of existing methods.

To study urban change from a VUA perspective it is necessary to detect change at scale throughout the entire city, whilst taking into account that the urban landscape differs tremendously between cities. This domain shift between cities is a challenge for existing USCD methods as they generally focus on *discrete change*, such as a building being constructed (Huang et al. 2024), which requires extensive manual labeling and an a priori definition of what is considered change. To overcome these limitations we propose a self-supervised learning approach that learns to detect urban change from unlabeled panoramic images.

To enable self-supervised learning at city-scale we built the AC-1M (Amsterdam Change) dataset of over 1.1M images, the largest USCD dataset by a significant margin. Furthermore, we propose EMPLACE, a self-supervised approach using an adaptive triplet loss that learns local change features without the need for labelling procedures while being robust to fleeting changes such as cars, people, and lighting. We evaluate its efficacy as a pre-training method for linear fine-tuning and zero-shot, and find we outperform SOTA models in both settings. By using EMPLACE we overcome the need for costly labelling processes thereby allowing our model to capture the full extent of change across the urban landscape. Our contributions are as follows:

1. We build the AC-1M, the largest Urban Scene Change Detection dataset to date containing over 1.1M panoramas curated to be within a single meter of each other.
2. We introduce tEMPoraL urbAn Change lEarning (EMPLACE), a self-supervised method for learning change detection features robust to noise such as weather, people, and cars. We demonstrate EMPLACE's potential in a zero-shot setting and as a pre-training method.
3. In a case study of Amsterdam we show that EMPLACE can find visual elements of change without an a priori definition of what constitutes it. We uncover both small and large visual elements and show that the size of visual change correlates differently with housing prices, demonstrating that visual change is indicative of socio-economic variation across cities.

Related Work

Visual Urban Analytics

The field of Visual Urban Analytics has a myriad of studies that have shown that neighbourhood visuals provide insight into socio-economic indicators such as mean income (Suel et al. 2019), housing prices (Law, Paige, and Russell 2019) or voting patterns (Gebru et al. 2017), as well as human perception such as liveliness (Dubey et al. 2016), scenicness (Seresinhe, Preis, and Moat 2017), uniqueness (Ordonez and Berg 2014), or perceived safety (Naik et al. 2014). While the predictive capability of these methods is solid, they use black box techniques and as such cannot uncover specific visual elements that influence socio-economic inequity (Alpherts, van Noord, and Ghebreab 2023). More useful examples such as trash detection (Sukel, Rudinac, and Worring 2020), pothole detection (Ma et al. 2022; Koch and Brilakis 2011), or quantifying greenery (Seiferling et al. 2017) exist, but they are supervised: Using a predefined notion of what is considered an important element. By approaching VUA through USCD we could get closer to uncovering new visual elements that potentially influence the condition of a neighbourhood.

Urban Scene Change Detection

The current field of USCD revolves around a supervised and *bi-temporal* paradigm: datasets consist of *two* images per location, taken before and after a change, which have been labelled with a segmentation map where change has taken place (Varghese et al. 2018; Zhao et al. 2019; Chen, Yang, and Stiefelhagen 2021; Lei et al. 2021). Commonly used datasets include: The PCD, consisting of 200 images of tsunami-damaged areas in japan, and the VL-CMU-CD (Alcantarilla et al. 2016), consisting of 1362 images used in autonomous vehicle navigation. Both are too narrow in scope and too small for self-supervised learning.

Furthermore the PCD, or its successor the PSCD (Sakurada 2018), are bi-temporal with aligned panoramas. In a real-world environment, over multiple years, panoramas are often misaligned due to in-the-wild differences in driving patterns of the capturing vehicle. As we are sampling our images from a real world distribution, our panorama distribution also has this misalignment requiring our method to be robust to the resulting visual noise. Finally, while panorama datasets for Amsterdam have been created before, they only consist of a single image per location and as such do not fit our task (Ibrahimi et al. 2021; Yildiz et al. 2022).

The CityPulse dataset is the only USCD dataset with multiple images per location consisting of 4465 square viewpoint images of places with building permits. However, this dataset is still bi-temporal and supervised: consisting of pairs of images with a binary label for whether a building has been constructed or not. We refer to this notion of binary prediction as *discrete* change prediction.

Attempts at steering away from supervision exist: Weakly supervised training has been employed in (Sakurada 2018) but this is an extrapolation of an existing labelled dataset. Self-supervised pre-training has been employed by (Ramkumar, Arani, and Zonooz 2022) but revolves around training on augmented images from a supervised dataset. We contribute a large scale (1.1M) unsupervised dataset with multiple images per location, as well as a self-supervised method for learning change features without labels.

Method

Our goal is to learn to detect arbitrary change between two images taken at the same location in an unsupervised manner. We hypothesize that when presented with three images taken at the same location, on average, images closer in time will exhibit less change than images further away. As such we introduce EMPLACE, a self-supervised learning method for learning visual representations using an adaptive

triplet loss. In the following, we describe the construction of the training dataset, the mining of triplets, and the adaptive triplet loss. Lastly we will also describe our method for evaluating our model's capability to detect change.

Construction of the AC-1M Dataset

Given a time series at location i of n panoramic images $p^{(i)} = (p_1^{(i)}, p_2^{(i)}, ..., p_n^{(i)})$ such that $p_k^{(i)}$ corresponds to timestamp $t_k^{(i)}$, we aim to capture change in a self-supervised manner following the principle that on average $p_a^{(i)}$ and $p_b^{(i)}$ should show less change than $p_a^{(i)}$ and $p_c^{(i)}$ for $t_b^{(i)} - t_a^{(i)} < t_c^{(i)} - t_a^{(i)}$. As such we build a dataset of clusters of three or more panoramas taken at the same location. Currently, no large-scale tri-temporal change detection dataset exists for this purpose, thus we created a new dataset called the AC-1M: A dataset of panorama clusters taken within a metre of each other, we show an example in Figure 2.

The AC-1M is constructed through a clustering and selection procedure applied to the Amsterdam Panorama Database: a collection of 6M panoramas taken in Amsterdam from 2016 to 2022. The steps taken to extract clusters are as follows:

1. The city is divided into polygons of its 386 neighbourhoods to reduce the computational complexity of clustering. All polygons are dilated by five metres to ensure clusters on the polygon edge are included.
2. For each polygon, the GPS coordinates of all panoramas within it are retrieved and candidate clusters are calculated using the DBSCAN (Ester et al. 1996) algorithm with a radius of 1 metre and the minimum amount of samples for a cluster to be considered set at $n = 3$.
3. For each candidate cluster, the cluster centre is used to retrieve all panoramas within a radius of 1 metre. This to ensure no images appear in multiple clusters. Clusters on water are excluded as well as clusters with multiple height values (due to tunnels).
4. Post-processing is applied by discarding the bottom 800 pixels to remove the capture vehicle. This results in panoramic images of 4000 x 1200. The heading parameter is then used to rotate all panoramas in the same direction after which a black rectangle is placed to block out the car antenna at the front and back of the vehicle, as can be seen in Figure 2. As the capturing vehicle was replaced over the years the antenna may otherwise induce a spurious correlation when predicting change.

The initial collection of 6M panoramas is reduced to **1.1M** panoramic images of 4000 x 1200 pixels assigned to **254k** clusters forming the **AC-1M**. This is an order of magnitude larger than existing datasets for USCD ($\leq$5000 images). A visualisation of the clusters plotted on a map of Amsterdam is shown in Figure A.1 in the Appendix. A comparison of datasets is shown in Table 1. The average image per cluster is 4.29 and the median images per cluster is 4. Both the polygons and the panoramas are available from the Amsterdam Municipality API. The details on how to retrieve the AC-1M will be available at github.com/Timalph/EMPLACE.

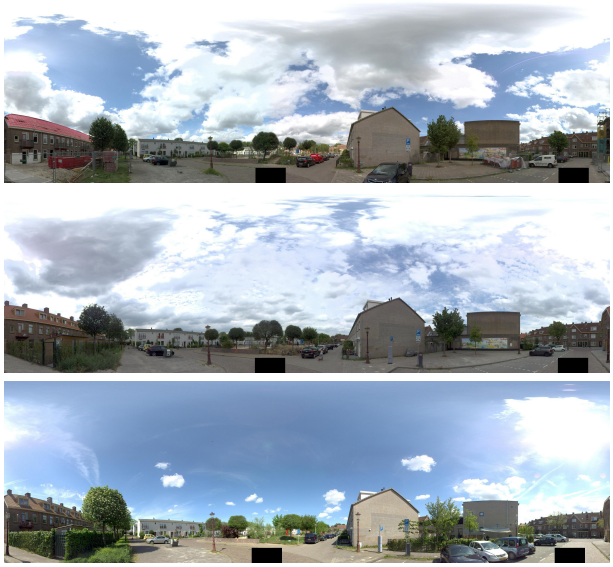


Figure 2: Example of a cluster from the AC-1M. From top to bottom images taken on 08-08-2016, 30-07-2018, 12-05-2022. Subtle changes happen over time such as the redoing of the roof on the left, the building of the fence on the right, and the chopping of the tree in the middle.

Dataset	# Images	# Locations
TSUNAMI	200	100
PSCD	1540	770
CityPulse	4465	371
AC-1M (Ours)	**1087047**	**254911**
AMS-Buildings (Ours)	**2354**	**404**
AMS-Trees (Ours)	**1374**	**273**

Table 1: Comparison of USCD datasets.

Triplet Mining and Loss

To train on the AC-1M we use a triplet loss in combination with three images: an anchor, a positive, and a negative image, where the anchor and positive image are closer together in time than the anchor and negative image. Note that positive and negative describe the distance in time to the anchor, and not whether labelled change is present in the images. This allows the model to learn visual change in a self-supervised way, relying on the temporal difference as the training signal. We mine triplets from our set of clusters where a triplet consists of three images where:

$$\Delta_{AP} < \Delta_{AN} \tag{1}$$

$$\text{where } \Delta_{AP} = t_{pos}^{(i)} - t_{anc}^{(i)} \tag{2}$$

$$\Delta_{AN} = t_{neg}^{(i)} - t_{anc}^{(i)} \tag{3}$$

$$\Delta_{PN} = t_{neg}^{(i)} - t_{pos}^{(i)} \tag{4}$$

which results in the amount of triplets that can be mined from a cluster of size n being $\binom{n}{3}$. Note that for all triplets:

$$t_{anc}^{(i)} < t_{pos}^{(i)} < t_{neg}^{(i)}$$

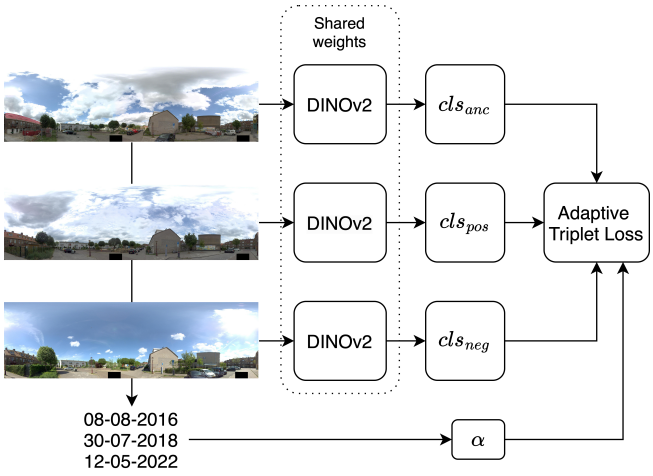


Figure 3: Overview of the model architecture. Image triplets perform a forward pass through a Siamese backbone with DINOv2 weights to calculate the cls tokens. The image dates are used to calculate the margin α. The cls tokens and margin α are used to calculate the adaptive triplet loss.

The full set of possible mined triplets is 2 million. For our purposes these triplets are filtered along certain constraints to steer them towards capturing meaningful changes. For example, by restricting the anchor and positive image to be within thirty days of each other, and the anchor and negative image to be more than three years apart. Note that this method allows the model to naturally become robust to changes in weather and lighting conditions.

To train using these triplets we use an adaptive triplet loss as shown in Eq. 5 where the margin α is defined in Eq. 6. We use a scaling margin as the observed changes are not linear with respect to time, as sudden changes can occur in between images. For images taken close together we often find that there is no change, whereas if there are multiple years in between images there may be many large changes. Yet, this process of change is typically not gradual but rather happened as a sudden jump in between two images.

$$\mathcal{L}(p_{anc}^{(i)}, p_{pos}^{(i)}, p_{neg}^{(i)}) = max(||f(p_{pos}^{(i)}) - f(p_{anc}^{(i)})||_2 - ||f(p_{neg}^{(i)}) - f(p_{anc}^{(i)})||_2 + \alpha, 0) \quad (5)$$

$$\alpha = \begin{cases} 0.5 * \left(\frac{\Delta_{PN}}{365}\right)^2 & \text{if } \Delta_{PN} < 365 \\ \frac{\Delta_{PN}}{365} - 0.5 & \text{else} \end{cases} \quad (6)$$

Model

For our transformer model we use ViT-B/14 initialised with DINOv2 weights (Oquab et al. 2024) as it has been shown to outperform other methods by a comfortable margin on the task of USCD (Huang et al. 2024). Our images are downsized to 700x210 for analysis, which results in the use of 14x14 patches that divide the image into a sequence length of 751, including the cls token. An overview of the model architecture is shown in Fig 3.

	Δ_{AP}	Δ_{AN}	# of triplets
SI-1	$1 < x < 31$	$375 < x$	14361
SI-2	$275 < x < 475$	$750 < x$	77125
SI-3	$275 < x < 475$	$1125 < x$	36384
SI-4	$275 < x < 475$	$1500 < x$	13344

Table 2: Distances in days between the anchor and positive image, and anchor and negative image for different SI setups.

Figure 4: An example of cut-and-flip data augmentation.

During training we introduce a new data augmentation step we call cut-and-flip augmentation: Panoramas differ from regular images in that they are circular. They consist of images taken in four directions of a capturing vehicle that are projected to a visual space where the horizontal axis of the image is circular and periodic. We enforce this visual nature through cut-and-flip augmentation where given a triplet, a random vertical cut is made and the cut images are swapped around. An example of cut-and-flip is shown in Figure 4.

To conclude, each training iteration consists of: 1) a forward pass of a triplet where the triplet loss is calculated applied to the euclidean distances of the resulting cls tokens, 2) a forward pass with the cut-and-flip augmented triplet, and 3) a backward pass performed using the cumulative loss.

Experiments and Evaluation

In this section we will describe our procedure for training on the AC-1M, the parameters, and our evaluation procedure through order prediction. We will then describe our method for discrete change prediction, the models we evaluate against, and the creation of two *new* discrete change detection datasets: AMS-Buildings and AMS-Trees.

Training EMPLACE

We randomly split the AC-1M dataset into a training, validation, and test set using a 70/20/10 split. This split is performed by cluster, which means every cluster only appears in one set. We decide on triplet constraints by calculating the SI, or Sampling Interval, which is 375 days. This corresponds to the median time between images in clusters of the AC-1M. We use the SI to evaluate four training setups: SI-1, SI-2, SI-3, SI-4 where the number is used to describe Δ_{AN} in SI. For our setups Δ_{AP} is between 275 and 475 days, except for SI-1 in which it is between 1 and 31 days. The lower bound of Δ_{AP} is always more than 1 day to ensure the task

(a) AMS-Buildings (b) AMS-Trees

Figure 5: Examples of cluster and image pairs from AMS-Trees and AMS-Buildings.

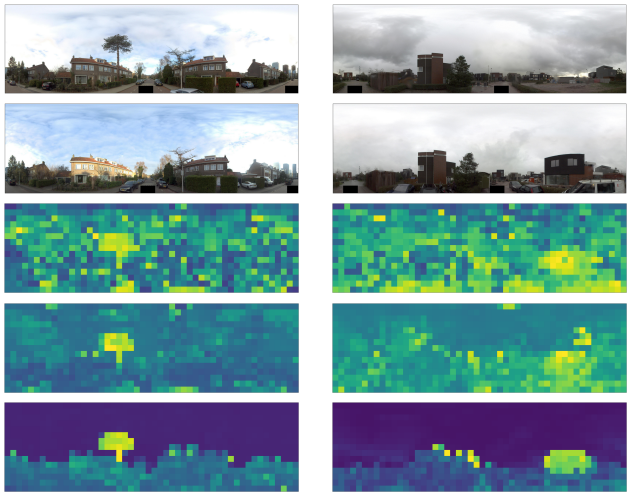


Figure 6: Heatmap comparison. Top two rows show the images between which the heatmaps are computed. Row three is ImageNet, row four is after training for discrete change, row five is zero-shot EMPLACE.

Training Setup \ Test set	SI-1	SI-2	SI-3	SI-4	All
SI-1	**.975**	.683	.726	.742	.744
SI-2	.959	**.903**	**.948**	.978	**.936**
SI-3	.926	.834	.937	**.981**	.900
SI-4	.895	.771	.846	.952	.840
✗	.709	.591	.602	.624	.616

Table 3: Results for EMPLACE on the task of order prediction, with the training setup in the form of SI-X on the y-axis, and the test set on the x-axis. The last column is the cumulative score on all test sets normalized by test set size. The last row shows the performance without training, or a ViT-B/14 model with DINOv2 features.

does not become too easy. These training setups, alongside the number of triplets after filtering are shown in Table 2. For each SI setup we train a ViT-B/14 using the adaptive triplet loss and cut-and-flip augmentation. We use the Adam optimizer with a learning rate set to 10^{-5}, a batch size of 64, and grad norm set to ≤ 0.5. Training and evaluation was conducted on 4 NVIDIA GeForce GTX 1080 Ti GPUs.

Order Prediction

To evaluate model accuracy and capability to detect change on the validation and test set we introduce the task of *order prediction*. Where the model is presented with a triplet and has to place the positive and negative image in the right temporal order based on euclidean distance to the anchor. We utilize early stopping if the accuracy has not improved for five epochs. While the validation set consists only of triplets using the same SI setup as the training set, after training we also evaluate every SI training setup on the test sets of other SI setups as well as on all test sets combined.

Evaluating Discrete Change

After training our EMPLACE model in a self-supervised manner, we also evaluate the performance of EMPLACE on *discrete* change prediction as introduced by (Huang et al. 2024). In this setting the model is presented with two viewpoint images and is tasked to predict whether change has occurred. This task is *supervised* and as such we will describe the construction of two labelled datasets for this purpose, the linear head necessary to turn the cls prediction into a discrete output, fine-tuning setup, training parameters, and backbones we test alongside EMPLACE.

AMS-buildings and AMS-Trees

To evaluate EMPLACE on the task of discrete change prediction we construct two labelled datasets using the same method as (Huang et al. 2024), the SOTA method for evaluating discrete change prediction: AMS-Buildings and AMS-Trees. Both datasets are constructed from the AC-1M test set to ensure the images are not seen by EMPLACE during training. AMS-Buildings contains 2327 image pairs *with* change and 1815 without. AMS-Trees contains 1157 with change, and 1084 without. Examples of images from both datasets are visible in Figure 5, while comparisons to existing datasets are shown in Table 1. Both datasets will be available at github.com/Timalph/EMPLACE.

Training setup We test the performance of EMPLACE as a pre-training method on the AMS-Buildings and AMS-trees by splitting both datasets into train, validation, and test sets following a 70/20/10 split and fine-tune the EMPLACE model that scores best on order prediction to perform discrete change prediction. We use a Siamese network with EMPLACE as the twin backbone model to compute the cls tokens of an image pair and concatenate them into a final layer to transform the input into a scalar prediction as shown in Eq. 3 in (Huang et al. 2024). We assess the performance of EMPLACE by comparing it to three pre-trained backbone models: ResNet101 (He et al. 2015), DINOv2 (Oquab et al. 2024), and CLIP (Radford et al. 2021). We fine-tune all models on both the AMS-Buildings and AMS-Trees. We use the Adam optimizer with a learning rate set to 10^{-5}, a batch

Buildings					
Model	Pre-Training	Acc	Prec	Rec	F1
ResNet101	✗	$.647 \pm .013$	$.709 \pm .007$	**.851** $\pm .029$	$.772 \pm .011$
CLIP	✗	$.706 \pm .023$	$.74 \pm .027$	$.775 \pm .026$	$.755 \pm .019$
DINOv2	✗	$.701 \pm .023$	$.764 \pm .02$	$.798 \pm .027$	$.778 \pm .014$
EMPLACE	SI-2	$.732 \pm .031$	$.802 \pm .04$	$.792 \pm .031$	**.794** $\pm .016$
EMPLACE zero-shot	SI-2	$.761 \pm .025$	$.792 \pm .027$	$.718 \pm .074$	$.75 \pm .037$
EMPLACE zero-shot	SI-4	**.781** $\pm .022$	**.840** $\pm .031$	$.703 \pm .058$	$.763 \pm .031$
Trees					
Model	Pre-Training	Acc	Prec	Rec	F1
ResNet101	✗	$.705 \pm .046$	$.600 \pm .111$	$.819 \pm .067$	$.676 \pm .071$
CLIP	✗	$.74 \pm .037$	$.686 \pm .124$	$.759 \pm .083$	$.7 \pm .058$
DINOv2	✗	$.762 \pm .028$	$.758 \pm .116$	$.739 \pm .088$	$.732 \pm .051$
EMPLACE	SI-2	**.855** $\pm .003$	**.853** $\pm .007$	**.924** $\pm .007$	**.885** $\pm .001$
EMPLACE zero-shot	SI-2	$.765 \pm .033$	$.803 \pm .062$	$.752 \pm .068$	$.773 \pm .038$
EMPLACE zero-shot	SI-4	$.764 \pm .026$	$.799 \pm .029$	$.718 \pm .073$	$.753 \pm .036$

Table 4: Results on AMS-Buildings and AMS-Trees. Best scores are shown in bold.

size of 16, grad norm set to ≤ 0.5, and early stopping after not having improved for 3 epochs. Training and evaluation was conducted on 1 NVIDIA GeForce GTX 1080 Ti GPU. We report the means and standard deviations of 10 runs with different seeds on the test set.

Visualising Change Detections

As our EMPLACE model has learned features for change detection during training we can compute a heatmap by using the euclidean distance between the token output of the Vision Transformer for zero-shot change prediction. An example of these heatmaps is shown in Figure 6. Our method for zero-shot change prediction therefore consists of moving a window over the image, and calculating the mean value of the euclidean distances in the patches of the window. If anywhere on the heatmap this mean value is above a certain threshold we output a detected change, and output no detection if otherwise. The optimal threshold and window size are both learned solely on the validation set.

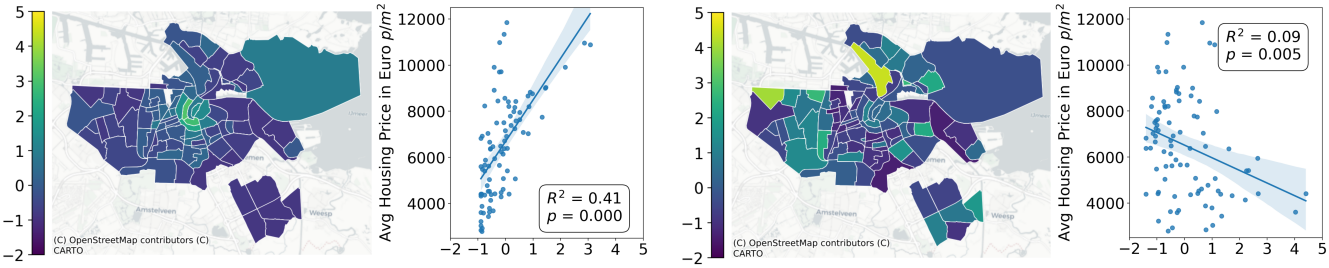
Results

Order Prediction

Order prediction is the task of putting two images in the correct temporal order with respect to an anchor image. As such, we consider this a simple evaluation method to evaluate the change detection properties of USCD models. The results for the order prediction task are shown in Table 3. We see that SI-2 EMPLACE (EMPLACE trained on SI-2) scores best overall by quite a large margin. Presumably this is because SI-2 EMPLACE has the hardest setup, relatively having the closest temporal distance between positive and negative images. The performance of SI-2 EMPLACE on the SI-3 test set, as well as high scores on the SI-1 and SI-4 test sets also indicate that SI-2 EMPLACE is most robust to visual noise present in urban images, and as such has learnt visual change most effectively. We see that while SI-1 EMPLACE outperforms the other models on its own test set, this performance does not generalize to other test sets indicating that the training objective of SI-1 EMPLACE is too easy: in SI-1 EMPLACE both anchor and positive image are taken during the same season, which is not the case for the other setups. Another interesting result is that SI-4 underperforms SI-2 and SI-3 on its own test set, seemingly indicating that the training set of SI-4 suffers from being too small; The stricter our triplet constraints are, the smaller the training set will be. Lastly, we see that this task is not trivial, as without training the model scores only .616 across all test sets.

Discrete Change Evaluation

The results of the discrete change evaluation experiments on the AMS-Buildings and AMS-Trees are shown in Table 4. We see that EMPLACE pre-training outperforms vanilla backbones when fine-tuned on both AMS-Buildings and AMS-Trees. We also see that the margins are higher on the AMS-Trees dataset. We assume this is due to change regarding trees being much more present in the AC-1M, while building construction is more scarce. We also observe that EMPLACE zero-shot actually outperforms the Siamese backbone on accuracy and precision on the AMS-Buildings. Furthermore, EMPLACE SI-4 zero-shot performs better than EMPLACE SI-2, potentially indicating that a larger Δ_{AN} forces the model to rely more on the built environment. The fine-tuning results indicate that EMPLACE is able to learn domain specific features about the Amsterdam that increase its performance on change detection tasks. The zero-shot performance indicates that the change detection task is also adequately doable without supervision. Lastly, we perform two additional studies: the first is an ablation study to evaluate the utility of the adaptive triplet loss and cut-and-flip augmentation, the results are shown in Appendix Table A.1. The second is a study on whether our method's performance is biased towards locations that show more change, the results of this are shown in Appendix Table A.2.



(a) σ of Detected Small Changes per Neighbourhoods

(b) σ of Detected Large Changes per Neighbourhoods

Figure 7: Detected small and large changes per neighbourhood. On the left the small changes are concentrated in the city center and show a positive correlation with housing prices. On the right the large changes are mostly found in the outskirts around large building projects in the North and West, showing a negative trend when plotted against housing prices.

Case Study

In addition to model comparison we also perform a case study to determine what visual elements of change exists throughout the full urban landscape of Amsterdam. We show how we can uncover different types of change without a priori definition of what constitutes relevant change. Finally, we distinguish between large and small visual change and show how this type of visual change correlates differently with a socio-economic indicator in the form of housing prices, illustrating how citizens of varying socio-economic backgrounds are potentially influenced by change in different ways.

Change Detection Setup

We run EMPLACE SI-2 on the entire Amsterdam test set and perform zero-shot change detection on the heatmaps on all pairwise comparisons in each cluster. The detection threshold above which we consider the token distance to constitute change is taken from the zero-shot experiments on AMS-buildings and AMS-Trees: We use the window size that performed best for the zero-shot runs for SI-2 in Table 4, which is 8x8. Of the 10 runs for which the zero-shot score was calculated, we take the maximum threshold value. We perform pairwise comparisons on all clusters in the AC-1M test set, and restrict our detections to 1 per image pair. This results in 889 detections on the 25k clusters of the AC-1M test set.

For small visual change, we cannot simply reduce the window size as the detection could then also be part of a larger change. As such we consider small visual change to be captured in 2x2 tokens, and not spill over to other neighbouring tokens. To capture these small changes, we perform a convolution operation to ensure the average of the tokens in a window is above the threshold value, while also being at least 120% larger than each of the surrounding tokens. Note that due to perpendicularity the tokens on the left and right side of the images wrap around. This results in 35177 detections on the 25k clusters of the AC-1M test set. Examples of retrieved small and large changes can be found in the Appendix in Figure A.2 and Figure A.3 respectively.

Correlation with Socio-Economic Indicators

To link the found changes to a socio-economic indicator we use the housing price dataset of (Groenen, Rudinac, and Worring 2022). The housing prices are available in a bin system for each "wijk" a cluster of neighbourhoods, which gives us 98 datapoints. We aggregate the changes per neighbourhood and show them in Figure 7. We observe that for small changes, there is a positive correlation with housing prices with an R^2 of .41 and a p value near 0. For large changes, we observe a negative trend. Because the R^2 is 0.09, we can not point to a negative correlation. However, the fact that the same correlation that holds for small changes does not hold for large changes points to the fact citizens of neighbourhoods with differing socio-economic demographics are confronted with different types of visual change. While small change happens mostly in the center of the city, large changes exist in the outskirts, where housing projects are being built on a continuous basis.

Conclusion

Our goal was to detect change throughout the city of Amsterdam. We built three datasets including AC-1M, the large USCD dataset to date, and built EMPLACE, a self-supervised method to learn strong change detection features. We introduced a triplet loss, cut-and-flip data augmentation, and an evaluation method in the form of order prediction and showed that EMPLACE was able to outperform SOTA methods for discrete change detection as a pretraining method for both buildings as well as trees. Additionally, we showed that EMPLACE was able to generate more distinct change heatmaps and that zero-shot change prediction scored high enough to detect various urban changes within Amsterdam.

In addition, we performed change detection on the entirety of Amsterdam, uncovered large and small visual changes, and showed that, in Amsterdam, expensive neighbourhoods are more likely to experience small visual change, while the inverse is true for large visual changes. We believe our work paves the way for a direction of USCD, and VUA by extent, that tries to uncover visual changes without an a priori definition of what to look for and can therefore be used more directly to investigate urban environments.

References

Alcantarilla, P. F.; Stent, S.; Ros, G.; Arroyo, R.; and Gherardi, R. 2016. Street-view change detection with deconvolutional networks. *Robotics: Science and Systems*, 12.

Alpherts, T.; van Noord, N.; and Ghebreab, S. 2023. Explaining Neighbourhood Liveability with Computer Vision: A Comparison of Methods on Usability and Practicality. Technical report.

Batty, M. 2019. Urban analytics defined. *Environment and Planning B: Urban Analytics and City Science*, 46(3): 403–405.

Chen, S.; Yang, K.; and Stiefelhagen, R. 2021. DR-TANet: Dynamic Receptive Temporal Attention Network for Street Scene Change Detection. arXiv:2103.00879.

Dubey, A.; Nikhil, N.; Parikh, D.; Raskar, R.; and Hidalgo, C. A. 2016. Deep Learning the City: Quantifying Urban Perception at a Global Scale. *Eccv*, 3: 398–413.

Ester, M.; Kriegel, H.-P.; Sander, J.; and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, 226–231. AAAI Press.

Gebru, T.; Krause, J.; Wang, Y.; Chen, D.; Deng, J.; Aiden, E. L.; and Fei-Fei, L. 2017. Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the United States. *Proceedings of the National Academy of Sciences of the United States of America*, 114(50): 13108–13113.

Groenen, I.; Rudinac, S.; and Worring, M. 2022. PanorAMS: Automatic Annotation for Detecting Objects in Urban Context. arXiv:2208.14295.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Huang, T.; Wu, Z.; Wu, J.; Hwang, J.; and Rajagopal, R. 2024. CityPulse: Fine-Grained Assessment of Urban Change with Street View Time Series. arXiv:2401.01107.

Ibrahimi, S.; van Noord, N.; Alpherts, T.; and Worring, M. 2021. Inside Out Visual Place Recognition. arXiv:2111.13546.

Joglekar, S.; Quercia, D.; Redi, M.; Aiello, L. M.; Kauer, T.; and Sastry, N. 2020. Facelift: A transparent deep learning framework to beautify urban scenes. *Royal Society Open Science*, 7(1).

Koch, C.; and Brilakis, I. 2011. Pothole detection in asphalt pavement images. *Advanced Engineering Informatics*, 25(3): 507–515.

Law, S.; Paige, B.; and Russell, C. 2019. Take a look around: Using street view and satellite images to estimate house prices. *ACM Transactions on Intelligent Systems and Technology*, 10(5): 1–19.

Lei, Y.; Peng, D.; Zhang, P.; Ke, Q.; and Li, H. 2021. Hierarchical Paired Channel Fusion Network for Street Scene Change Detection. *IEEE Transactions on Image Processing*, 30: 55–67.

Li, X.; Zhang, C.; Li, W.; Kuzovkina, Y. A.; and Weiner, D. 2015. Who lives in greener neighborhoods? The distribution of street greenery and its association with residents' socioeconomic conditions in Hartford, Connecticut, USA. *Urban Forestry & Urban Greening*, 14(4): 751–759.

Ma, N.; Fan, J.; Wang, W.; Wu, J.; Jiang, Y.; Xie, L.; and Fan, R. 2022. Computer Vision for Road Imaging and Pothole Detection: A State-of-the-Art Review of Systems and Algorithms. 1–16.

Muller, E.; Gemmell, E.; Choudhury, I.; Nathvani, R.; Metzler, A. B.; Bennett, J.; Denton, E.; Flaxman, S.; and Ezzati, M. 2022. *City-Wide Perceptions of Neighbourhood Quality using Street View Images*, volume 1. Association for Computing Machinery.

Naik, N.; Kominers, S. D.; Raskar, R.; Glaeser, E. L.; and Hidalgo, C. A. 2017. Computer vision uncovers predictors of physical urban change. *Proceedings of the National Academy of Sciences of the United States of America*, 114(29): 7571–7576.

Naik, N.; Philipoom, J.; Raskar, R.; and Hidalgo, C. 2014. Streetscore-predicting the perceived safety of one million streetscapes. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, (January): 793–799.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Assran, M.; Ballas, N.; Galuba, W.; Howes, R.; Huang, P.-Y.; Li, S.-W.; Misra, I.; Rabbat, M.; Sharma, V.; Synnaeve, G.; Xu, H.; Jegou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2024. DINOv2: Learning Robust Visual Features without Supervision. arXiv:2304.07193.

Ordonez, V.; and Berg, T. L. 2014. Learning high-level judgments of urban perception. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8694 LNCS(PART 6): 494–510.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Ramkumar, V. R. T.; Arani, E.; and Zonooz, B. 2022. Differencing based Self-supervised pretraining for Scene Change Detection. (11): 1–13.

Sakurada, K. 2018. Weakly Supervised Silhouette-based Semantic Change Detection.

Sakurada, K.; and Okatani, T. 2015. Change Detection from a Street Image Pair using CNN Features and Superpixel Segmentation. 61.1–61.12.

Seiferling, I.; Naik, N.; Ratti, C.; and Proulx, R. 2017. Green streets: Quantifying and mapping urban trees with street-level imagery and computer vision. *Landscape and Urban Planning*, 165(4): 93–101.

Seresinhe, C. I.; Preis, T.; and Moat, H. S. 2017. Using deep learning to quantify the beauty of outdoor places. *Royal Society Open Science*, 4(7).

Suel, E.; Bhatt, S.; Brauer, M.; Flaxman, S.; and Ezzati, M. 2021. Multimodal deep learning from satellite and street-level imagery for measuring income, overcrowding, and environmental deprivation in urban areas. *Remote Sensing of Environment*, 257(June 2020): 112339.

Suel, E.; Polak, J. W.; Bennett, J. E.; and Ezzati, M. 2019. Measuring social, environmental and health inequalities using deep learning and street imagery. *Scientific Reports*, 9(1): 1–10.

Sukel, M.; Rudinac, S.; and Worring, M. 2020. Urban object detection kit: A system for collection and analysis of street-level imagery. *ICMR 2020 - Proceedings of the 2020 International Conference on Multimedia Retrieval*, 509–516.

Varghese, A.; Gubbi, J.; Ramaswamy, A.; and Balamuralidhar, P. 2018. ChangeNet: A Deep Learning Architecture for Visual Change Detection.pdf.

Yildiz, B.; Khademi, S.; Siebes, R. M.; and van Gemert, J. 2022. AmsterTime: A Visual Place Recognition Benchmark Dataset for Severe Domain Shift. arXiv:2203.16291.

Zhao, X.; Li, H.; Wang, R.; Zheng, C.; and Shi, S. 2019. Street-view Change Detection via Siamese Encoder-decoder Structured Convolutional Neural Networks. *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 5(Visigrapp): 525–532.