Detecting Change for Multi-View, Long-Term Surface Inspection

Simon Stent[1]
sistent@cantab.net
Riccardo Gherardi[2]
riccardo.gherardi@crl.toshiba.co.uk
Björn Stenger[2]
bjorn.stenger@crl.toshiba.co.uk
Roberto Cipolla[1]
rc10001@cam.ac.uk

[1] Machine Intelligence Lab
Department of Engineering
University of Cambridge
Cambridge, U.K.

[2] Toshiba Research Europe Ltd.
Cambridge, U.K.

Abstract

We describe a system for the detection of changes in multiple views of a tunnel surface. From data gathered by a robotic inspection rig, we use a structure-from-motion pipeline to build panoramas of the surface and register images from different time instances. Reliably detecting changes such as hairline cracks, water ingress and other surface damage between the registered images is a challenging problem: achieving the best possible performance for a given set of data requires sub-pixel precision and careful modelling of the noise sources. The task is further complicated by factors such as unavoidable registration error and changes in image sensors, capture settings and lighting.

Our contribution is a novel approach to change detection using a two-channel convolutional neural network. The network accepts pairs of approximately registered image patches taken at different times and classifies them to detect anomalous changes. To train the network, we take advantage of synthetically generated training examples and the homogeneity of the tunnel surfaces to eliminate most of the manual labelling effort. We evaluate our method on field data gathered from a live tunnel over several months, demonstrating it to outperform existing approaches from recent literature and industrial practice.

1 Introduction

We address the problem of change detection between pairs of images taken at different times by a moving camera. Our motivation is the development of a non-contact inspection system, summarised in fig. 1, to be used for detecting anomalous visual changes on surfaces, and in particular tunnel linings. This application is of increasing social importance as our infrastructure ages and requires more efficient maintenance than existing, frequently labour-intensive, methods can provide. The problem is challenging for several reasons:

i) **Size and nature of changes.** Changes of interest are often small and subtle – e.g. a fattening in the width of a hairline crack or a patch of discolouration caused by organic

Pages 127.1-127.12
DOI: https://dx.doi.org/10.5244/C.29.127

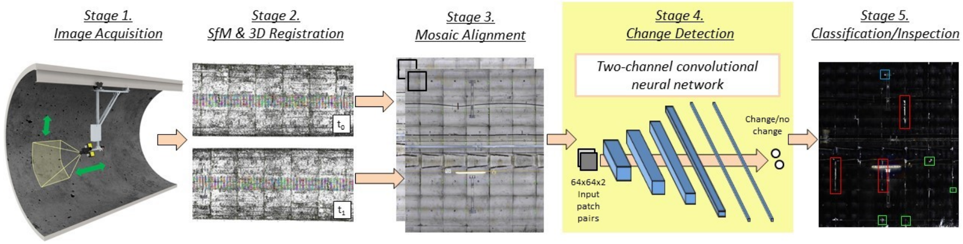


Figure 1: Overview of our machine vision system. The main focus of this paper is stage 4, in which changes are detected between registered sets of image mosaics captured at different times. We propose and evaluate a new approach to change detection using a two-channel convolutional neural network. The network learns a model for normal modes of image variation, so as to detect abnormal changes with fewer false positives. Sections 3 and 5 outline the system and proposed change detection method in more detail.

growth or concrete spalling. This property emerges from the nature of the change detection problem: as the period over which change is measured decreases, any algorithm is pushed against the intrinsic limits set by image resolution and sensor noise. In the datasets examined here, fewer than 0.07% of the pixels were labelled as changes of interest, and in a different scenario the ratio could be several orders of magnitude lower. Furthermore, while certain changes such as cracks are known in advance and can be explicitly detected (e.g. [20]), others may be too infrequent for explicit modelling and only detectable as anomalous to natural modes of image variation.

ii) **Nuisance factors.** A sizeable proportion of the observed change over time is caused by nuisance factors, either internal to the acquisition system (such as different image sensors, capture settings or lighting setup) or due to external causes (for example, seasonal changes of temperature and humidity). While tunnels are relatively static in comparison to other environments such as outdoor scenes, external conditions such as humidity and dust levels can cause sufficient variation in visual appearance to shroud more important structural changes of interest. Fig. 2(b) illustrates the variation in appearance from a random set of corresponding unchanged image patches taken at different times and conditions.

iii) **Registration error.** Achieving the pixel-accurate registration required for change detection is challenging because neither the sensor position nor the tunnel geometry can be reliably determined. Inaccurate or un-modelled geometry causes parallax errors when images are re-projected; in addition, a blanket change across the scene – caused for example by a change in tunnel humidity level – can make feature-based registration of any single image impossible.

Our system circumvents the need for improving both the registration and insensitivity to nuisance sources through machine learning. We train a two-channel convolutional neural network (CNN) which takes as input a pair of image patches and returns a measure of dissimilarity or change. CNNs have recently been shown to be very effective at learning invariance to certain modes of image variability. They require however large amounts of labelled image data. We have unlimited access to negative pairs (i.e. patches where no abnormal change

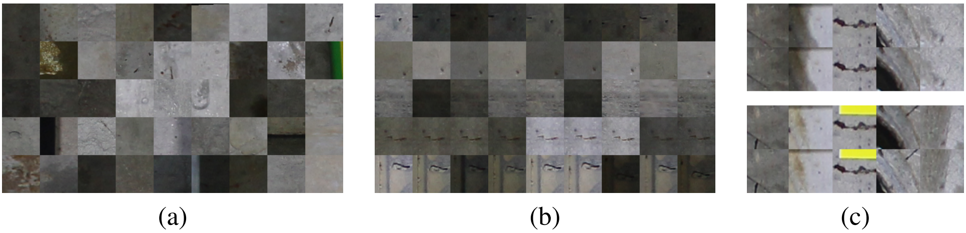


Figure 2: (a) Array of randomly sampled 64 × 64 pixel patches from the dataset. (b) The same as (a), but each row contains 9 different viewpoints of the same unchanged patch to illustrate the natural image variation and registration error. (c) Examples of changed patches; top rows are different viewpoints from t_r, bottom rows from t_q. (Best viewed on screen.)

has occurred) by taking registered viewpoints from different cameras from the same time. We supplement this with a smaller dataset of negative pairs across the different test times from regions where no changes of interest have occurred. This requires a limited effort in coarsely labelling a small subset of the test data. Together, these negative pairs capture much of the natural nuisance variation from lighting, registration errors and camera pose variation. For the positive (changed) pair generation, we provide randomly sampled pairs as well as synthetically generated changes. The homogeneity of the tunnel environment – illustrated by fig. 2(a) – allows a network to generalize well from a manageable amount of labelled ground-truth.

We evaluate our system using three sets of data from a live tunnel captured at different times. A trained inspector was tasked with simulating real changes in the tunnel between captures and a set of ground truth change images were generated for testing. We compare against an implementation of the current state of the art [12] and against the results of a manual inspection carried out by a second trained inspector in the field. The latter is of particular importance to industry, as it is commonly still the method of choice for tunnel inspection. To our knowledge, this is the first comparison of this kind reported in literature.

2 Background

We first define the problem of change detection for multi-view surface inspection. Given a reference image I_r and a query image I_q taken of a surface from different positions and under different imaging conditions at times t_r and t_q respectively, we seek a binary change mask, C, which is 1 at every position in I_q that has undergone a change of interest and 0 elsewhere. In practice, we assume that the two images have been registered into a common 2D coordinate frame using a surface model of the scene, acquired in our case via surface fitting on geometry recovered from Structure-from-Motion (SfM) as in [12].

The problem of change detection is then to determine:

$$P(C(\mathbf{p})=1\,|\,I_r(\mathbf{p}),I_q(\mathbf{p}))=f(I_r(\mathbf{p}),I_q(\mathbf{p})) \quad (1)$$

for any pixel or patch of pixels $\mathbf{p}$. The function f is a measure of change between the two image patches and can either be designed using domain knowledge or learned from a given dataset. The definition of change is always problem-specific; in our application we seek local changes in the state of the surface such as cracks, water ingress, rust and surface damage.

2.1 Related Work

Image-based change detection is an important part of many vision pipelines [8] with applications ranging from video surveillance and medical imaging to remote sensing and urban and environmental change detection. Before changes can be detected, images are typically first preprocessed to register them geometrically and correct for any radiometric variation [8]. In some situations, these steps can be achieved with little error. For example, in remote sensing, where the goal is to detect environmental changes such as the extent of deforestation on a 2D map built from satellite images, parallax effects are negligible and synthetic aperture radar is frequently used to lessen the effect of atmospheric and lighting change across time [3]. Many change detection methods thus assume pixel-accurate registration as a starting point.

In other situations, including our own, pixel-accurate registration is more difficult to achieve. In urban change detection for example, camera pose, geometry and radiometric variation are often quite severe [7]. Most methods use 3D scene geometry recovered from SfM and multi-view stereo for image registration, and focus only on detecting 3D changes such as the appearance or disappearance of urban structures [10, 13, 15]. The recent work of [5] aims instead to discover textural changes on planes in the scene, but their focus is on more significant variations (e.g. the change in appearance of a billboard) that can be temporally clustered and their method requires a comparatively denser sampling of the environment both in time and space. Mesh models recovered from SfM are used in [1] to register underwater images and detect changes in images of a coral reef after radiometric correction; this application however does not demand the detection of fine-grained changes.

While our system also relies on a geometric model for approximate registration, we sidestep the need for finer registration or radiometric correction by using a CNN, trained to detect unnatural changes between pairs of coarsely registered image patches. The idea of learning similarity functions f to match pairs of image patches has been approached in the past [6] and recent efforts to do so with CNNs have shown much promise. In [16], a CNN is trained to compute the stereo matching cost of 9×9 pixel patches, leading to state-of-the-art results on the KITTI stereo benchmark. In [21], several architectures are investigated for learning f between larger 64×64 pixel patches. They show good results for various matching tasks using a two channel network; we adopt a similar architecture but unlike their approach train directly on a mixture of task and synthetic data, for the inverse problem of detecting change rather than measuring similarity. Using task data for training allows us to learn invariance to task-specific variation rather than making prior assumptions (such as the distribution of translations due to registration error). The injection of synthetic data ensures that our network can remain sensitive to a specific important class of variation (fine cracks). Furthermore, unlike [21], we do not incorporate additional patches from larger scales in separate input channels since by design all of our patch pairs have similar sizes (corresponding approximately to 20×20mm).

Transfer learning, where the lower layers of networks trained for a task such as ImageNet classification [2, 11] are reused as generic image representations to solve other tasks, has been shown to be a successful strategy in a variety of computer vision problems [9]. The key benefits versus learning from scratch are that it circumvents the need for large task-specific labelled sets and long training times. We eschew this approach because: (i) the statistics of our dataset are likely to be very different compared to those of problems such as ImageNet, and (ii) we seek to control more tightly the modes of variation against which the network learns invariance.

Finally, in the field of structural monitoring, numerous tunnel inspection systems are in active development such as [4, 14] but to our knowledge there has been no other effort to detect general visual appearance changes in such scenes. We improve upon our previous work [12], by introducing a more robust and scalable capture, reconstruction and registration system (section 3) and removing the dependency on pixel-accurate registration required by our previously proposed change detection method (section 5).

3 System Description

In this section, we briefly outline the main steps and improvements in our system (fig. 1) leading up to the change detection stage.

Image Capture. In stage 1, overlapping 360 degree rings of images were gathered by an autonomous calibrated camera system running along a monorail. The camera system uses polarised lighting and orthogonally polarised lens filters to remove or attenuate image variation modes due to scene specularities. This was found to dramatically increase the number of image correspondences, particularly in wet areas of the tunnel where reconstruction was otherwise observed to fail.

Reconstruction and registration. Images from different times were processed independently via Structure-from-Motion (SfM) to return sparse point clouds (side views shown) and camera pose estimates in stage 2. The data was processed in overlapping parallel subsets corresponding to approximately 3 metre long sections. We used VisualSfM [19] for 3D reconstruction with accelerated SIFT features for matching [18] and added intermediate ring-closure checks to ensure complete reconstructions. To avoid the high computational cost of repeatedly running bundle adjustment on large sets of images, rings of images were treated independently given their immediate neighbouring rings, providing both efficiency and robustness during reconstruction. Neighbouring reconstructed subsets were registered across time in a piece-wise rigid fashion, using a similarity transform estimated via Procrustes alignment on a subset of confident feature correspondences. Unlike the single-image registration method of [12], this global alignment on a large set of images ensured that single images could still be successfully registered even in the presence of large changes in appearance.

Mosaicing for visualisation. A surface model was then estimated for each reconstructed subset from t_r. The same cylindrical assumption was used as in [12]. Unlike [12], points lying close to the surface were projected directly onto the surface and individual camera poses were refined (resectioned) to reduce registration error. Image mosaics were obtained by projecting constituent images onto the surface model and blending. This resulted in ghosting artefacts for areas which were off-surface but otherwise produced results which were sufficiently accurate for the visualisation of pixel-wide (0.3mm wide) surface cracks.

4 Datasets

We gathered and processed field data to produce two different test datasets, following the schedule detailed in fig. 3. Artificial changes such as cracks, leaks, rust and stickers were applied to the tunnel surface before the capture of $\mathcal{I}_{t_1}$ and $\mathcal{I}_{t_2}$. Some examples are shown in

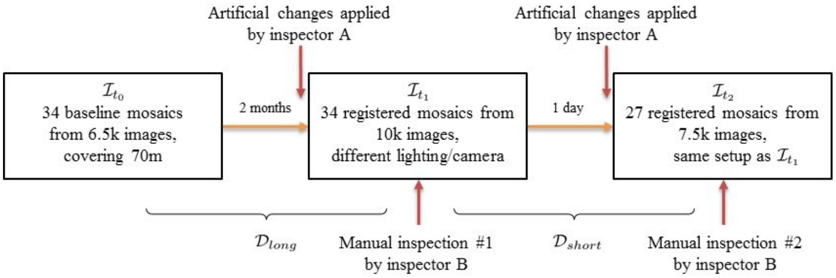

Artificial changes applied by inspector A
Artificial changes applied by inspector A
$\mathcal{I}_{t_0}$
34 baseline mosaics from 6.5k images, covering 70m
2 months
$\mathcal{I}_{t_1}$
34 registered mosaics from 10k images, different lighting/camera
1 day
$\mathcal{I}_{t_2}$
27 registered mosaics from 7.5k images, same setup as $\mathcal{I}_{t_1}$
$\mathcal{D}_{long}$
Manual inspection #1 by inspector B
$\mathcal{D}_{short}$
Manual inspection #2 by inspector B

Figure 3: Timeline and datasets gathered for the inspection experiments.

fig. 2(c). The changes were applied by a professional inspector and designed to be as realistic as possible. 90 changes were applied in total (45 in each instance), covering altogether less than 0.07% of all mosaicked pixels in the test set.

The resulting change detection datasets, $\mathcal{D}_{long}$ and $\mathcal{D}_{short}$, contain changes over two months and one day respectively. The one day dataset, $\mathcal{D}_{short}$, is more amenable to automatic change detection since within a shorter time frame the chance for new defects to appear (other than those purposely introduced as part of the test protocol) is lower. The changes applied in this instance were subtle and harder to detect for human observers, including variations in crack width and length. $\mathcal{D}_{long}$ is a more challenging dataset, using a different camera and lighting setup and more realistic temporal change of over two months. The changes here also include the appearance of new cracks, objects or defects.

Manual inspections were carried out by a second professional inspector before each capture of $\mathcal{I}_{t_1}$ and $\mathcal{I}_{t_2}$. The inspector was informed of what kind of changes to be aware of before each test, and during the second inspection was allowed to consult with his own notes from the first.

5 Change Detection Method

For change detection, we generated a set of reference and query image pairs by dividing the mosaic images into 64 × 64 pixel patches, and then for each patch projecting only the image from the nearest camera. Doing so achieved two goals: firstly, within each block the patches were free from compositing artefacts and secondly we avoided the computational cost required to process all the available overlapping image pairs independently.

5.1 CNN Architecture

Our CNN architecture was similar to the two-channel approach proposed by [21], consisting of four convolution layers of depths 32, 64, 128 and 512, and two fully connected layers of depth 512, with a softmax layer to classify the input pair between changed and unchanged states. The first three convolution layers are followed by 2 × 2 max pooling, and all hidden layers are followed by a non-saturating Rectified Linear Unit (ReLU) non-linearity as in [2]. The input is two-channel, with the first layer of 7 × 7 × 2 pixel filters operating directly on both 64 × 64 pixel gray-scale patch inputs normalised to have zero mean and unit variance. The results of [21] suggest that this is preferable in practice to maintaining separation of the

	Training Set	Number of Pairs	Positive Pair Generation Method
(i)	TS-R	250,000	Random
(ii)	TS-SR	250,000	Semi-random
(iii)	TS-C	250,000	Crack
(iv)	TS-SM	250,000	Small mixture, $\frac{1}{3}$ from each of (i-iii)
(v)	TS-LM	1,000,000	Large mixture (i)+(ii)+(iii)+(iv)

Table 1: CNN training sets used. (i-iv) compare the effect of different positive pair generation methods; (v) compares the effect of training set size vs (iv).

channels until a later layer; one likely reason for this is that high-frequency information can be immediately compared between the patches, providing valuable similarity information that might be otherwise lost through pooling.

5.2 Synthetic Crack Generation

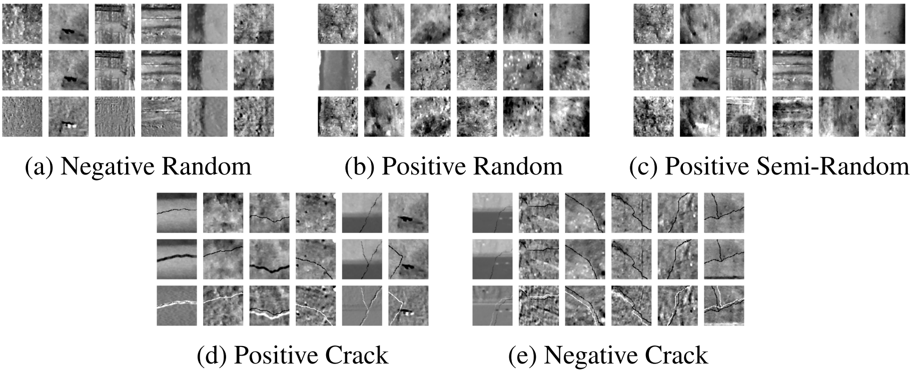
We generated synthetic crack images for training by blending real image patches with a crack mask. Each mask was created by randomly sampling a small set of crack support points within a region encompassing the image patch. A minimum spanning tree was formed over the support points, and branches from the tree recursively subdivided to generate new support points, each of which was perturbed randomly according to a pre-generated perlin noise map. The resulting crack map was rasterised, with width determined by a second perlin noise map, resulting in a realistic random crack image generator. The perlin maps provided coherent randomness in the crack behaviour across image scales.

5.3 CNN Training

Taking a single corresponding pair of mosaic images from t_r and t_q as a training set, we trained four separate networks with the architecture described in section 5.1 from random initialisations, each using one of the training sets (i,ii,iv and v) from table 1. The training sets were split equally into positive (changed) and negative (unchanged) samples, with negative samples reused across training sets (i, ii and iv) for fairness of comparison, and to gauge the effect of using different strategies for positive pair sampling on the network's performance.

Fig. 4 illustrates various sets of training pairs and their differences. To generate each column of negative (unchanged) pairs in (a), we sample a random location and draw two overlapping image patches from each of the t_r and t_q image datasets. Ground truth is required to avoid sampling locations which have changed; to create it, the training mosaic is assigned coarse labels, which are collected into a discrete change mask.

To generate each positive pair in (b) a new random location is chosen in each of the t_r and t_q image datasets and patches are extracted. The semi-random patches in (c) take half of the random patches from (b) and half of the negative patches from (a), thus ensuring that a positive sample is tied to every negative sample in the dataset. Finally (d) and (e) are generated using the synthetic crack generator described in section 5.2. We either take an image pair from (a) and add a crack to one of the pair, or use a single base image which we arbitrarily translate to generate two patches. The translation is drawn from a uniform distribution over ±7 pixels in x and y, empirically accounting for the majority of surface



(a) Negative Random (b) Positive Random (c) Positive Semi-Random

(d) Positive Crack (e) Negative Crack

Figure 4: Sample training pairs (rows 1+2) and their difference images (row 3) from different training sets: (a) negative (unchanged) pairs; (b) positive (changed) random pairs, with both members chosen randomly (TS-R); (c) semi-random positive pairs, combining (a) and (b) (TS-SR); (d) positive crack pairs, including crack appearance/disappearance, extension and widening (TS-C); (e) negative crack pairs (TS-C). (Best viewed on screen.)

registration errors. The translation being known, we can modify the crack appearance in either of the images to simulate crack extension or widening.

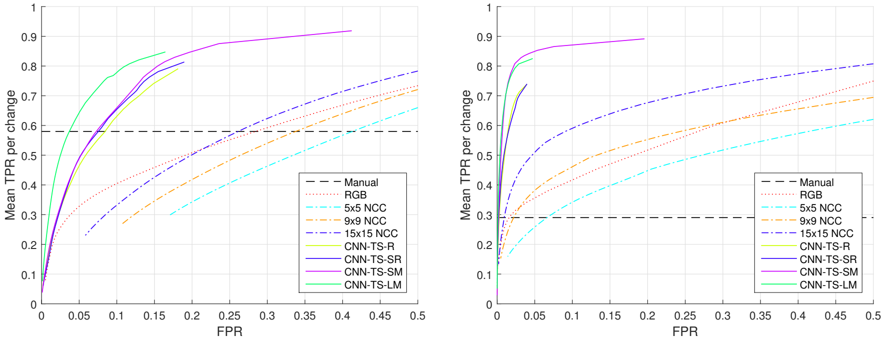
Each network was trained identically until convergence of a log loss cost function on the softmax output. We trained using stochastic gradient descent, using a batch size of 512 image pairs, a momentum of 0.9 and a weight decay of 0.0005. We applied 50% dropout in the two fully connected layers to reduce overfitting. Similarly to [2], we initialised layer biases preceding ReLU layers with the constant 1, to accelerate early learning by providing the ReLUs with positive inputs. Filter weights were initialised by sampling from a zero-mean Gaussian distribution with standard deviation 0.01. Convergence was observed within 30 training epochs in all cases. The networks were implemented in `MatConvNet` [17] with `CuDNNv2` support.

6 Evaluation and Discussion

We compared our method against both the manual inspection results and a version of [12] modified to run on our high-resolution test datasets. In all methods, we employed the same geometric prior as [12], which restricts change detection to segments of the image that lie on the tunnel surface.

6.1 Quantitative Evaluation

Fig. 5 illustrates change detection performance over the two test datasets. The x-axis represents the False Positive Rate (FPR), the proportion of actual negatives which are incorrectly assigned as positive. The y-axis shows the average ratio of pixels in each ground truth change that were correctly labelled as having changed. This metric was chosen in order to fairly represent all changes and to be fair to the human inspector, since the distribution of the area of changes is broad – from very small and thin cracks to large leaks. `Manual` refers to the manual inspection by a trained inspector, which uncovered 29% of changes in $\mathcal{D}_{short}$ and



(a) (b)

Figure 5: Change detection performance of the various methods compared for (a) $\mathcal{D}_{long}$ and (b) $\mathcal{D}_{short}$ datasets. Our proposed CNN approach outperforms existing automated methods on both datasets.

58% in $\mathcal{D}_{long}$; RGB shows the performance of pixel-to-pixel absolute differencing; and the method of [12] is applied using NCC windows of varying sizes from 5×5 to 15×15 pixels.

In both datasets, we see that our CNN approach, even when trained in a naive manner, outperforms the existing methods by a significant margin. RGB and NCC methods both require good registration, which is not equally reliable throughout the datasets – especially in $\mathcal{D}_{long}$, where the capture setup varied significantly. While the manual method outperforms ours at very low FPR, it is not possible to retrospectively trade off FPR for TPR so the performance is bounded below what CNN can achieve in theory.

Among the CNN methods, the performance difference between training with random or semi-random positive pairs is negligible (CNN−TS−R vs. CNN−TS−SR), but performance can be seen to improve when the data is augmented with synthetic crack data (CNN−TS−SM). This is especially true of $\mathcal{D}_{short}$, where 27% of changes involve cracks expanding or extending (vs 0% in $\mathcal{D}_{long}$). Increasing the size of the training set (from CNN−TS−SM to CNN−TS−LM) improves performance significantly in $\mathcal{D}_{long}$ but has little effect in $\mathcal{D}_{short}$. One possible explanation is that $\mathcal{D}_{long}$, which was captured over a longer time period and with a different capture setup, contains more nuisance variation and thus benefits from a larger training set to learn from.

Table 2 shows the percentage of detected changes at different FPR thresholds for various methods. Here we define a detected change as one containing >50% of positive pixels. Our method shows significant improvement over [12] in both datasets and over manual inspection in $\mathcal{D}_{short}$, though manual inspection discovers more changes at very low FPR setting. It should be noted that not all false positives are strictly misclassifications; many correspond to real anomalous changes that were not part of the labelled changes of interest.

6.2 Qualitative Evaluation

Several more factors are noteworthy when comparing the tested approaches.

(i) ***Time required.*** The manual inspections took 70 minutes for $\mathcal{D}_{long}$ and 30 minutes for $\mathcal{D}_{short}$, with several additional hours required to process the results. While the automated processes were not run exclusively on the test datasets from end to end in a

Change detection dataset:	$\mathcal{D}_{short}$			$\mathcal{D}_{long}$		
FPR per pixel:	0.01	0.05	0.10	0.01	0.05	0.10
Manual inspection	0.29	0.29	0.29	**0.58**	0.58	0.58
Modified version of [12]	0.20	0.55	0.64	0.00	0.00	0.32
Ours: `CNN-TS-LM`	**0.73**	**0.87**	**0.87**	0.26	**0.65**	**0.84**

Table 2: Percentage of artificial changes detected by the compared systems at different false positive rates. Changes are considered detected if they are greater than >50% positively labelled.

single stream, processing them would take an order of magnitude extra time on a single desktop machine without employing significant parallelisation.

(ii) ***Objectivity.*** Despite the cost and time for processing, the automated approach has numerous advantages - the foremost being that it is completely objective. Our system does not suffer from inattentional blindness and can view every point in the tunnel at the same resolution.

(iii) ***Scalability.*** The performance of the automated approach scales favourably with data size, as fig. 5(b) demonstrates. Manual inspection performance drops with scale, due to human fatigue over a repetitive task.

(iv) ***Visualisation.*** Automation allows data to be visualised at any later date. In contrast, manual inspection notes are gathered by hand and typed up to computer and are difficult to cross-reference across time.

(v) ***Manual advantages.*** Finally, it should be noted that manual inspection can pick up other changes that our current system cannot detect, such as defects occluded behind cables and small geometric defects in tunnel ring alignments.

7 Conclusions and Future Work

In this paper we have presented a system for the detection of changes from multiple views of a tunnel surface. We proposed a novel approach to change detection using a two-channel convolutional neural network and demonstrated its favourable performance on field data versus competing solutions.

Our approach can be straightforwardly adapted to different textured surfaces and novel scenarios with minimal manual training effort - this is one avenue for future work. It is also very efficient for processing data on the scale of a working system, where there may be kilometres of data to survey.

Another area for future work is the classification of changes after change detection. It may be feasible to re-use the same CNN for this task, with a few changes to the architecture and training procedure. A further extension is to extend the synthesis of positive changes beyond cracks to other common defects such as rust and water ingress.

Acknowledgements. We thank Mark Farmer at National Grid, David Naylor at Mott MacDonald and Cédric Girerd, Peter Long and Kenichi Soga at the University of Cambridge for their help and cooperation in the project.

References

[1] O. Delaunoy, N. Gracias, and R. Garcia. Towards detecting changes in underwater image sequences. In *OCEANS 2008-MTS/IEEE Kobe Techno-Ocean*, 2008.

[2] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in NIPS*, pages 1097–1105, 2012.

[3] J. Liu, M. Gong, J. Zhao, H. Li, and L. Jiao. Difference representation learning using stacked restricted boltzmann machines for change detection in SAR images. *Soft Computing*, pages 1–13, 2014.

[4] K. Loupos, A. Amditis, C. Stentoumis, P. Chrobocinski, J. Victores, M. Wietek, P. Panetsos, A. Roncaglia, S. Camarinopoulos, V. Kalidromitis, D. Bairaktaris, N. Komodakis, and R. Lopez. Robotic intelligent vision and control for tunnel inspection and evaluation - the ROBINSPECT EC project. In *Robotic and Sensors Environments (ROSE), 2014 IEEE International Symposium on*, pages 72–77. IEEE, 2014.

[5] K. Matzen and N. Snavely. Scene chronology. In *ECCV*, pages 615–630, 2014.

[6] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *CVPR*, pages 1–8, 2007.

[7] T. Pollard and J.L. Mundy. Change Detection in a 3-D World. In *CVPR*, June 2007. ISBN 1-4244-1179-3.

[8] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Trans. Image Process.*, 14(3):294–307, 2005.

[9] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *CVPRW*, pages 512–519. IEEE, 2014.

[10] K. Sakurada, T. Okatani, and K. Deguchi. Detecting changes in 3D structure of a scene from multi-view images captured by a vehicle-mounted camera. In *CVPR*, 2013.

[11] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[12] S. Stent, R. Gherardi, B. Stenger, K. Soga, and R. Cipolla. Visual change detection on tunnel linings. *Machine Vision and Applications*, pages 1–12, 2014. ISSN 0932-8092.

[13] A. Taneja, L. Ballan, and M. Pollefeys. Image based detection of geometric changes in urban environments. In *ICCV*, pages 2336–2343, 2011. ISBN 978-1-4577-1102-2.

[14] tCrack system by terra vermessungen ag, 2012. URL http://www.terra.ch/en/crack-detection.html.

[15] A.O. Ulusoy and J.L. Mundy. Image-based 4-D reconstruction using 3-D change detection. In *ECCV*, pages 31–45. Springer, 2014.

[16] J. Žbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, 2015.

[17] A. Vedaldi and K. Lenc. MatConvNet – convolutional neural networks for MATLAB. *CoRR*, abs/1412.4564, 2014.

[18] C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT), 2007. URL http://cs.unc.edu/~ccwu/siftgpu.

[19] C. Wu. VisualSFM: A Visual Structure from Motion system, 2011. URL http://homes.cs.washington.edu/~ccwu/vsfm/.

[20] T. Yamaguchi and S. Hashimoto. Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications*, 21:797–809, 2010.

[21] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015.