Byvshev, Petr; Pham, Truong; Xiao, Yu

Image-based Renovation Progress Inspection with Deep Siamese Networks

Published in:
Proceedings of the 2020 12th International Conference on Machine Learning and Computing, ICMLC 2020

DOI:
10.1145/3383972.3384036

Published: 15/02/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Byvshev, P., Pham, T., & Xiao, Y. (2020). Image-based Renovation Progress Inspection with Deep Siamese Networks. In *Proceedings of the 2020 12th International Conference on Machine Learning and Computing, ICMLC 2020* (pp. 96-104). ACM. https://doi.org/10.1145/3383972.3384036

Image-based Renovation Progress Inspection with Deep Siamese Networks

Petr Byvshev
Aalto University
Espoo, Finland
petr.byvshev@aalto.fi

Pham-An Truong
Aalto University
Espoo, Finland
truong.pham@aalto.fi

Yu Xiao
Aalto University
Espoo, Finland
yu.xiao@aalto.fi

ABSTRACT
Various specialized machine vision systems have been proposed for item inspection, robot supervision and quality control in manufacturing and construction industries. However, construction industries are still lacking solutions for automating the progress inspection in building renovation projects. As smartphones becoming increasingly pervasive among construction workers and use of smartphone photos for documentation getting more popular, in this work, we propose machine learning methods for automatic progress recognition in renovation projects from smartphone images. Renovation progress inspection is formulated as an ordinal classification problem, with every class representing one stage of the renovation process. The baseline solutions inspired by the popular deep learning architectures like VGG19, ResNet, Xception, DenseNet and MobileNet do not benefit from the temporal property of the data. Consecutive stages share a substantial amount of visual features, which makes it difficult to differentiate between them. To cope with that, we design special networks - Ordering Nets which utilize the consecutive property of the classes to predict the order of a photo pair. For the special use case, we train the 1-Step- Net to recognize progress from two subsequent photos. The extracted order information increases classification accuracy and provides more precise temporal prediction. We report our results on a new Reno-2018 dataset - a two-part collection of photos that covers bathroom and kitchen renovation steps. The applicability of our approach is demonstrated on a simulated progress estimation task. Our method significantly improves the temporal accuracy of stage predictions compared to the base deep neural network models.

CCS Concepts
• Computing methodologies→Visual inspection.

Keywords
Inspection; deep learning; progress estimation; ordinal classification.

1. INTRODUCTION
The increasing need for automated visual inspection in different industrial verticals is driving the development and adoption of machine vision techniques. According to the market analysis, more than 50% of the global machine vision market is held by the quality assurance and inspection segment 1 . Compared with other industrial verticals, construction sector lags behind in use of digital advancements and does not utilize robotization enough [1]. Taking building renovation projects as an example, quality control and inspection are vital to ensure that future users receive products in time and free from defects.

One important task of inspection is to track completeness status and recognize potential delays. Currently this labor is done manually: inspectors visit every facility, estimate the time to complete, mark incomplete work and defects, and then enter the information into a statistics managing software or a paper document. This procedure requires expensive man hours and lacks automation and digitalization. To solve this problem, a digital solution for automatically detecting progress from visual data (e.g. images) is highly demanded. Such a solution could reduce the costs of inspections and provide statistics for resources optimization.

Due to the lack of open datasets, that represent progress estimation, we created a labeled photoset - Reno-2018, through a collaboration with a construction company. It is composed of photos, taken during a series of bathroom and kitchen plumbing renovations and labeled with progress status by professionals. Each photo belongs to a single stage and contains visual information, specific for one of the renovation steps defined according to the common practice in the construction industry. The dataset is used for training and testing the deep learning methods.

We formulate the process of image-based renovation progress inspection as an ordinal classification problem. The existing deep models learn class specific features and do not utilize ordinal information of the classes - this leads to high temporal misclassification. Indeed, the visual properties of the consecutive classes can be shared, causing a high confusion rate. We designed a new deep learning framework to extract temporal relations between the classes. In the core of the visual analysis is a deep artificial neural network - the Ordering Net, which learns the chronological order of the classes. The special case is a 1-Step-Net - it is trained to recognize the visual changes in two subsequent stages. The extracted information increases classification accuracy and, more importantly, provides a more reliable progress estimate. To evaluate the system, we simulate a progress estimation task with a sequence of photos from a single apartment. Our solution is supposed to recognize whether a new input photo (or a batch of photos) comes from a new, significantly different renovation stage or belongs to the previously observed stage.

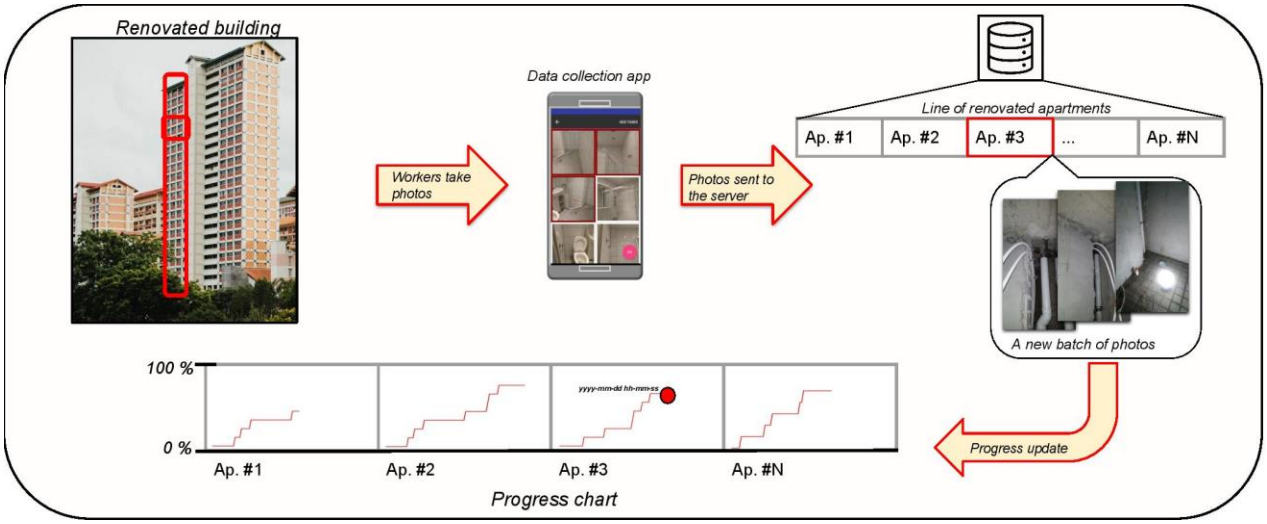

Figure 1. Progress inspection system.

A summary of our contributions including experimental results are listed below.

• We built Reno-2018, a set of photos taken during the process of bathroom and kitchen renovations project and labeled with the progress information (i.e. stage of the process). The dataset can help in designing and evaluating image-based methods for process inspection.

• We test the most popular CNN models to classify stages of plumbing renovation through minimally preprocessed photos with the help of the Reno-2018 dataset. These models show low temporal precision, when predicting the renovation stage.

• We analyze visual interclass properties of the Reno-2018 dataset based on extracted CNN. The results statistically demonstrate the visual similarity of consecutive classes.

• We propose Ordering Nets - deep Siamese networks [2], that directly address the temporal property of the renovation process. The learned representation improves the utility of image-based progress estimation, providing a more temporally-adjusted stage prediction.

• We apply a special variation of the Ordering Nets - the 1-Step-Net, that learns to recognize specific changes in the working environment. We develop an application strategy which helps to detect when a new renovation stage is initiated based on the incoming photo (or a batch of photos), that achieves 93.4% one-stage confidence interval.

The following parts of this paper are structured as below. In Sections 2 and 3 we briefly discuss the inspection system overview and related work. In Section 4 we describe the technical details and challenges of the Reno-2018 dataset. In Section 5 we formulate the progress inspection task as an image classification problem. In Section 6 we investigate image retrieval and deep learning methods for the renovation stage classification. Section 7 describes Ordering Nets and the 1-Stage-Net that explore the ordinal property of the stage classification problem. Section 10 discusses the results and encountered challenges before we conclude the work.

The proceedings are the records of the conference. ACM hopes to give these conference by-products a single, high-quality appearance. To do this, we ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download a template from [2], and replace the content with your own material.

2. RELATED WORK

In the literature, vision inspection has been applied in a variety of manufacturing applications, such as item inspection, robot supervision, production quality control [3] and many more. For instance, a context-aware CNN is demonstrated to detect surface defects in manufacturing processes and control for visual imperfections [4]. In [5], authors use photogrammetry - a method that estimates 3D geometry of a scene from a list of photos, for progress estimation of an excavation process.

Progress inspection can be formulated as an ordinal image classification problem, with each class representing one stage in the process. So far, the cases of ordinal image classification are dominated by health-related datasets, because of the consecutive nature of the disease phases [6, 7]. Such cases usually contain a large number of samples, reaching tens of thousands. However, these cases are often limited by low number of stages and unbalanced distributions of images across classes. Another case of ordinal classification is the photo-based age prediction, where for every photo a model is expected to infer the age range of the person [8]. In our case of renovation progress inspection, we are dealing with limited amount of data, and various levels of class differentiability.

Regarding progress estimation of indoor renovation projects, there are different strategies varying in cost and effectiveness. One is based on the information retrieval from the scene [9]. In [10], the authors proposed to evaluate the differences between query images and trained images based on template matching and object detection. A key challenge comes from the fact that the error rate of template matching and object detection increases with the complexity of indoor scene. Another approach is to estimate the state based on photogrammetry. The authors of [11] show that photogrammetric image processing can provide dimensions of indoor construction. With that information, the system can verify the quality and progress of construction work based on generated 3D models. However, the manual dimension extraction and marker setup increase the costs of this method. In [12], the authors propose to recognize the state of indoor renovation based on

decision trees. Although the accuracy is quite high, the approach lacks scalability, since it requires manually predefined decision trees. In contrast with previous works, this work aims to provide automatic and scalable renovation progress inspection based on smartphone photos, without extra hardware infrastructure or manually-defined decision trees. Our approach is ideologically similar to the template matching, but the key differences come from the fact that deep learning models do not require predefined templates and statistically learn typical features of process stages.

3. MOTIVATION AND OVERVIEW

Presently extensive amounts of visual data is collected for documentation of industrial processes. Construction companies collect photos of different stages of a renovation process, such photos often contain only the timestamps and the name of the project. In a nutshell, we are looking for a system that extracts the temporal logic from the photos of ordered classes to provide relevant statistics of a construction process, in our case - bathroom/kitchen renovation. One use scenario could be described as follows: a worker takes photos periodically (once a day), using a data collection mobile application, the photos are processed and relevant statistics is extracted into the monitoring tool, the site manager/inspector can remotely access the on-line progress monitoring tools and make necessary planning and management decisions. In the current work we propose the computer vision-based progress inspection system that provides sequential stage identification. The system would process every new batch of photos associated with specific attributes (for example: location, apartment number, date and time,...etc.) and output the progress update in real time (Fig. 1).

Additionally, such a system can utilize weakly-labeled photos to better learn the dynamics of visual changes observed in various construction processes.

4. STAGE CLASSIFICATION DATASET

Due to the lack of open datasets that cover renovation projects, we create a new dataset called Reno-2018 which can help design the image-based solutions for progress inspection in building renovation projects. The dataset consists of two parts: bathroom photos and kitchen photos, each covering a different renovation process. The bathroom part is composed of photos, taken from 7 bathrooms at 10 different renovation stages (Fig. 2). The kitchen part covers 8 renovation stages observed in 6 kitchens. All photos were taken during February-March of 2018. We developed an Android data collection application, it is designed for taking and labeling photos on-site. The user interface is illustrated in Fig. 3. Samsung SM-G920F mobile camera (resolution 5312×2988) and Sony E6653 (resolution 3840×2160) were used to take photos. The photos were labeled by construction specialists according to the internal renovation documentation norms.

Tables 1 and 2 illustrate the distribution of photos across the classes. Due to irregularities in the construction process and data collection, the distribution of samples across classes is not strictly uniform. To balance for that, we used common augmentation techniques, such as translation, rotation, change of luminance and adding noise; the augmentation factor for each class is not greater than 4. After the preprocessing, the dataset contains 3708 bathroom and 1486 kitchen images. We only use the bathroom part of the dataset to test the models, since it is more complete and balanced. The kitchen photoset is added to test the final progress estimation scenario.

Table 1. List of bathroom renovation stages and the respective number of taken photos.

Stage	Number of photos	%
Initial stage	128	3.5%
Demolition	244	6.6%
Plastering	617	16.9%
Plumbing	877	23.6%
Conduits	621	16.7%
Laying the concrete	286	7.7%
Water insulation	247	6.7%
Tiling	365	9.8%
Chrome piping	183	4.9%
Final cleaning	140	3.7%
Total	3708	100%

Table 2. List of kitchen renovation stages and the respective number of taken photos.

Stage	Number of photos	%
Demolition	199	14.5%
Cutting water	86	5.7%
Sewer and water lines	191	12.8%
Carpenter works	126	8.4%
Tiling	212	14.2%
Wall protections	266	17.9%
Electrical installations	252	16.9%
Elements reconstruction	164	11.0%
Total	1486	100%

Figure 2. Example photos of ten renovation stages.

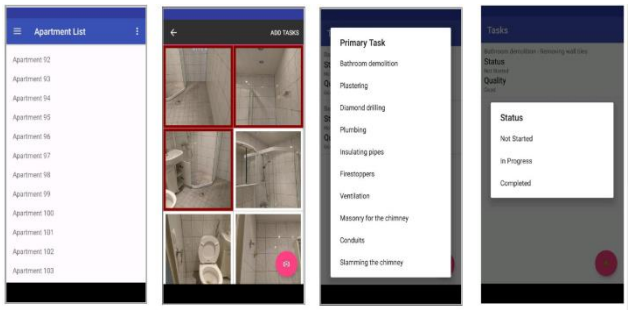

Figure 3. User interface of Android application for collecting data. (A) Apartment names list. (B) Grid of bathroom photos. (C) Task adding action. (D) Progress stage selection list.

To test the classification models, we randomly split the data into 85%- training and 15% - testing. The random split is adequate since all the rooms follow the same renovation process and have the same size. We see at least three challenging aspects of the datasets. First, the division of a continuous process into stages can be subjective, but such discretization is necessary, as the data collection was performed in batches. Second, a stage can contain visual features of the previous stage, for example, parts of the conduit system may be visible during the laying the concrete floor stage (See Fig. 2 for more examples). Third, due to the practical aspects of the renovation, the stages are not strictly ordered: some classes may include images with visual clues from a different temporally adjacent class.

5. SCENE CLASSIFICATION PROBLEM

Humans are generally very effective at understanding the contents of natural or industrial scenes. We can evaluate all the important details and aspects of the environment to provide a required description. But a simple task of figuring out if you are viewing a kitchen or bathroom is still highly challenging for computer vision. The renovation progress estimation (or progress estimation in general) is semantically even more challenging, as it requires a specialist, who visits the site and reports the current work status (i.e. stage of the renovation progress) based on his/her expertise. This approach is time- and labor-consuming. The delay in the process of information collection and aggregation also reduces the effectiveness of progress management during the renovation processes.

In this work we investigate deep learning based vision inspection techniques that can increase the level of automation in renovation progress inspection. Compared with other vision inspection systems such as quality control in assembly line, renovation progress inspection must take into account the ordinal structure of the data. As introduced in Section 4, a plumbing renovation process consists of a list of consecutive scenes/stages. Unlike regular classification, in ordinal classification confusing two distant or subsequent stages are fundamentally different errors, with the former leading to a greater error in the progress estimation.

Assuming that a renovation process consists of N stages, we can formulate two types of stage classification problems. The first problem is to recognize the current stage of the renovation project, given an image, for example, taken from a bathroom under renovation without prior knowledge of previous photos. In the second problem, given a set of ordered images, we need to provide an ordered list of stages, that covers these photos. The detailed design of methods that solve these tasks are described in sections 6 and 7. Evaluating the solutions, we need to consider not only the classification accuracy, but also the chronological agreement of predictions. A number of quality measures have been proposed to evaluate machine learning and statistical methods [13], and one such measure is Weighted Kappa Index (WKI), defined by Cohen to judge inter-rater agreement in an ordinal classification problem [14]. The level of agreement varies from 0 to 1 (see Table 3).

Table 3. Table for interpretation of Weighted Kappa, after Landis Koch (1977)

κ	Strength of agreement
<0.20	Poor
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Good
0.81-1.00	Very good

Weighted Kappa is used as an index to measure the performance of a classifier in a problem, where the categories have a logical ordering. The Weighted Kappa is defined as:

$$\kappa = 1 - \frac{\Sigma_{i,j} w_{i,j} O_{i,j}}{\Sigma_{i,j} w_{i,j} E_{i,j}}, \quad (1)$$

where N is the number of classes, i, j $\subset$ (1, 2, ..., N),$O_{i,j}$is the number of observations classified in the i-th category by the prediction model and they are in the j-th category in the correct classification (i.e. ”true value”), $E_{i,j}$is outer product between the two classification histogram vectors (prediction and ”true value”), normalized such that E and O have the same sum, $w_{i,j}$ is weight penalization for every pair i, j. We used quadratic penalization: $w = \frac{(i-j)^2}{(N-1)^2}$.

The Kappa index takes values from -1 to 1, where 1 means a perfect agreement between classifier and true value, 0 - random assignment and -1 would mean a perfect disagreement. It is reasonable to use the index in the Reno-2018 case, as for example, the classification of Stage 3 to be Stage 8 should be penalized more than Stage 3 to Stage 4.

6. STAGE RECOGNITION METHODS

6.1 CNNs for Stage Recognition

We propose to train a CNN for recognizing the corresponding stage from a single image. For the training, we followed the transfer learning ideology, meaning that low-level visual features obtained from large computer vision datasets can be reused for a new classification task. In image classification, it is a common practice to use the transfer learning method in order to save computational power and time. In that case, a network, trained on a large dataset (ImageNet, for example), is used as an initial state for the new model. This allows to skip learning simple features in the first convolutional layers of the network. For the stage identification, we test the most popular CNN architectures: DenseNet, VGG19, MobileNet, ResNet50 and Xception [15-19]. First convolutional layers of networks were fixed to reuse the stable features that were learned from the ImageNet dataset. The last convolutional layer is followed by a 512- units ReLU layer and a 10-units softmax layer for the final classification (8 units for the kitchen case). We tested two loss strategies: the usual cross-

entropy (CE) error and the fixed class (FC) squared-error [20]. The last one is indicated to be sensitive to class ordering:

$$L_{ce} = -\sum_{i=0}^{k-1} y_i \, log\big(f(x_i)\big) \quad (2) \qquad L_{fc} = (c - a^T f(x)_i)^2 \quad (3)$$

where L_{ce} is the cross-entropy error, L_{fc}- the fixed class squared-error, k - number of classes, y - target categorical class, f(x) - the output layer of the net with a softmax activation, c - numerical value of the target class, a - constant vector ($a_i = i$). For every input image the CE networks output a confidence vector, we take the maximum confidence argument as the class prediction. The FC networks output a continuous numerical value of a class and we take the closest integer as the prediction. All networks were trained until the validation accuracy converges: early stopping is applied if the accuracy does not improve for 25 epochs with a 250 epochs cap.

Table 4 compares different CNN architectures in the stage recognition task. VGG19 and MobileNet demonstrate similar performance with the latter one being a significantly lighter model. In the following experiments we will focus on the VGG19 architecture.

Table 4. Comparison of popular CNN architectures

Network	Size (Mb)	Acc. (CE)	Acc. (FC)	WKI (CE)	WKI (FC)
VGG19	549	43.2%	35.6%	0.64	0.73
DenseNet	33	35.4%	35.1%	0.55	0.62
MobileNet	16	41.3%	40.2%	0.64	0.71
ResNet50	98	23.6%	26.4%	0.51	0.64
Xception	88	41.2%	39.5%	0.62	0.69

Table 5 shows the confusion matrix for the VGG19 solution: every row represents the distribution of predictions for the respective class. The trained model shows 43.2% top-1 accuracy, 85.3% top-2, Kappa Index of 0.64, where top-k means taking k maximum confidence predictions. Such a big difference between top-1 and top-2 could be explained by the visual statistics of the classes in the dataset. The next section demonstrates visual relations of the classes and explains the behavior of the CNN solution.

Table 5. Confusion matrix for the VGG19 solution (%).

	0	1	2	3	4	5	6	7	8	9
0	100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	4.5	66.0	20.0	5.0	0.0	0.0	0.0	0.0	0.0	4.5
2	0.0	1.2	43.5	30.7	12.8	11.5	0.0	0.0	0.0	0.0
3	0.0	0.0	26.3	45.0	3.2	4.3	6.6	14.2	0.0	0.0
4	0.0	0.0	36.2	15.9	27.5	2.9	0.0	17.3	0.0	0.0
5	0.0	0.0	25.1	12.5	12.5	31.2	15.6	0.0	0.0	0.0
6	0.0	0.0	3.4	27.6	6.9	20.7	24.1	17.2	0.0	0.0
7	3.0	0.0	0.0	33.3	3.0	3.0	0.0	51.5	3.0	3.0
8	4.3	0.0	0.0	17.4	4.3	0.0	0.0	60.9	4.3	8.7
9	0.0	0.0	5.9	0.0	0.0	0.0	0.0	0.0	0.0	94.1

6.2 Interstage Dissimilarity

The CNN-based solution shows a significant difference between top-1 and top-2 accuracies. This could mean a strong intermix of consecutive classes/stages. To visualize that we examine the interstage similarity based on the CNN features.

To obtain the CNN-based similarity index of two stages the following procedure was held. First, we selected the last convolutional layer as the feature extractor. For every image this layer produces 512 feature maps. We take the maximum value of a feature map for every input, ignoring the spatial position of activation. This way we get 512 activations vectors for every sample in a class and every class is represented by a vector cluster. We suggest that distance between the centroids of two clusters describes the interclass visual similarity. Calculating that distance for every two classes gives us a similarity matrix. The distances are normalized to take values from 0 to 1, where 0 means full similarity and 1 - full differentiability of the classes (see Fig. 4).

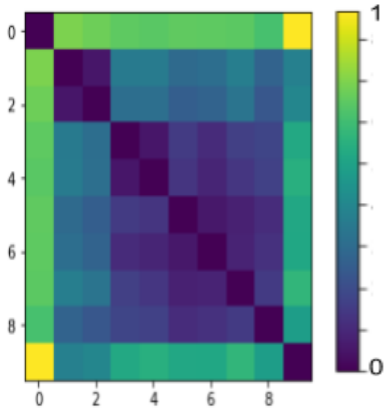

Figure 4. Normalized similarity matrices for 10 stages of Reno-2018 dataset based on CNN features. The matrix is symmetrical, each row represents the degrees of similarity between the respective class and all the classes. The heat map contains values from 0 (on the diagonals) - full similarity within a class, to 1 - full differentiability of two classes.

From the matrix we can see a reasonable pattern: closer stages are visually more similar and the difference increases for the stages that are more chronologically separated. For example, the matrix shows that stages 2 and 3, 4 and 5 are greatly intermixed, which can also be confirmed by a visual inspection in Fig. 2 ('Demolition' and 'Plastering', 'Plumbing' and 'Conduits' respectively). Similar pattern can be observed in the confusion matrix of the VGG19 − CE solution (see Table 5), where most of the misclassified samples belong to the neighboring stage.

7. ORDER ESTIMATION

While convolutional neural networks are considered to be the method to extract class specific features for image labeling, such architectures don't benefit from the ordinal property of the data. For the renovation progress estimation we need to incorporate the chronological nature of the images. A reasonable solution should learn the renovation flow and understand the order relations of the stages. Namely, the system should understand that certain visual features appear in a specific order and that the stage prediction has to take into account information about the previous stages. To achieve that, we deploy Siamese-network structure - a special kind of neural networks, designed to operate on pairs of images. A

Siamese network is a composition of two identical neural networks, the last layers of each are merged for the final classification. A Siamese-network is trained to recognize certain associations between the input pairs, such as, relative camera pose, geometrical or semantic similarity, specific object displacement [21-23].

7.1 Ordering Nets

A CNN-based solution does not explicitly use the ordinal property of the data. To learn the chronological relations of the stages we propose a specific Siamese-network - Ordering Net. To train the network we sample pairs of images from the training section of the Reno-2018 dataset, resulting in 10000 pairs. For every pair the only information we extract is the order - which photo comes from the earlier stage. This way the network was trained to solve

binary classification problem - natural order vs. flipped order. For the problem we used a Siamese-network with shared weights, by taking two copies of a CNN network, initializing them with weights learned in the classification task and truncating to the final convolutional layer (See Fig. 5). The output of the Ordering Net is '0' or '1'. '0' output means that the input pair is correctly ordered, '1' - swapped. We can obtain a class prediction for a test image with the Ordering Net by pairing the test image with a number of photos from every training class. The pairs from the same class should produce a larger variance in the prediction. Therefore, the prediction class should correspond to the set of training images that brings the most uncertainty to the Ordering Net prediction.

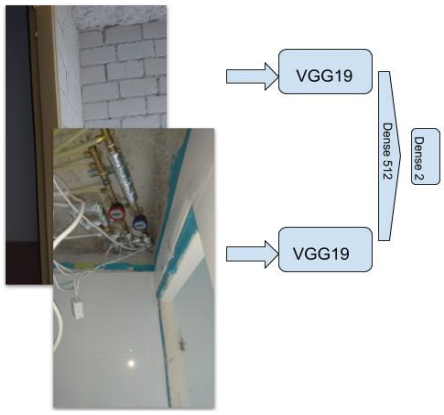

Figure 5. Example of an Ordering Net

Now, we can utilize this approach to correct the target stage prediction made by the CNN-CE or CNN-FC solutions. For every sample we take top-2 predictions and use the Ordering Net to finalize the prediction. It is similar to the ensemble methods [23] with an exception: the models provide structurally different information. The classification network uses visual features to provide class probabilities, while the Ordering Net gives relative position of an image in a sequence of stages. This method improves classification accuracy and significantly increases the Kappa index, meaning that the model provides a more informed prediction taking into account the order of the stages (see Table 6). The highest Kappa Index was achieved by combining prediction from VGG19 − FC and the Ordering Net.

Table 6. Accuracy and Kappa index for CNN net-work, Ordering Net and combined.

Bathrooms		
	Accuracy	Kappa Index
SIFT-based	27.3%	0.35
VGG19 − CE	41.2%	0.64
Ordering Net	38.4%	0.67
VGG19 − FC	35.6%	0.73
VGG19 − CE+ Ordering Net	48.4%	0.78
VGG19 − FC+ Ordering Net	39.1%	0.80
Kitchens		
SIFT-based	22.3%	0.34
VGG19 − FC	32.6%	0.70
Ordering Net	37.4%	0.66
VGG19 − CE	37.6%	0.72
VGG19 − CE+ Ordering Net	42.4%	0.77
VGG19 − FC+ Ordering Net	35.1%	0.76

8. SEQUENTIAL STAGE IDENTIFICATION

We showed that Ordering Nets can learn ordinal relations of the sequential classes. In the following we will describe how a model based on an Ordering Net can help us in a progress estimation task. In the application scenario the photos are usually taken in batches. Some - from a later stage, some from the current one. Using the timestamps is not enough for the progress estimation as different apartments may be at various states. We can formulate the progress estimation task as a problem of sequential stage identification based on the incoming photos. For every new incoming photo we need to resolve if enough progress has been made to call it a new stage or if the photo still belongs to the same stage. To solve this problem we use an Ordering Net, trained specifically on the consecutive pairs. For this task we uniformly sample pairs of photos from the same stage and pairs from subsequent stages. Therefore, we have the same binary classification problem as the ordering problem, only this time the solution should specialize on the Figure 6: Example of the progress (Y-Coordinate) estimation method for 100 samples (X-Coordinate). Blue line - true progress, red - estimated progress. Every jump represents transition into a new stage. one-step gradations. We call this network '1-Step-Net' . For every pair of photos the 1-Step-Net predicts if they come from the same stage or subsequent. The network outputs a value between 0 and 1, where '0' output means that a pair belongs to the same class, '1'- to subsequent classes. We should keep in mind that most of the time we have access to more than one photo of a time-fixed renovation stage. So our task is to update the stage when enough visible changes have been made.

To evaluate the stage recognition strategy we carry the following tests. The trained model is given a sequence of 100 previously unseen photos taken from the same location. We consider two processes - the bathroom and kitchen renovations. The number of photos per stage is unknown. This way we model the inconsistency of the data collection process. Our task is to estimate the progress across these 100 samples - time steps. The progress can be visualized as a ladder, and every new stage is associated with going one step up (Fig. 6 - blue).

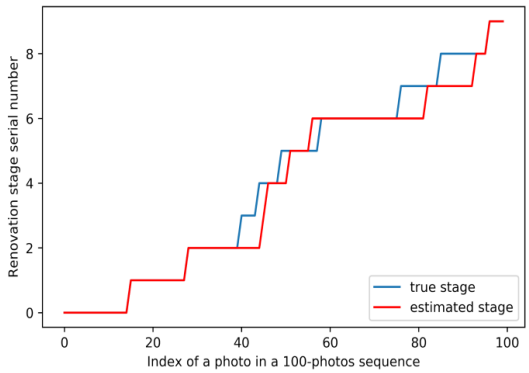

Figure 6. Example of the progress (Y-Coordinate) estimation method for 100 samples (X-Coordinate). Blue line - true progress, red - estimated progress. Every jump represents transition into a new stage.

We propose the following method for the real-time progress estimation:

1) Choose the sliding window size - s (we test window sizes of 2,4,6,8), that is a number of consecutive image pairs to feed into the 1-Step-Net. For simplicity, in the further explanation we will assume the window of size 4.

2) For every image j in a sequence feed 4 image pairs to the net: (j − 2, j), (j − 1, j), (j, j + 1), (j, j + 2).

3) For a sequence of N images the previous steps result in (N − s) × s matrix P of predictions.

4) A local ideal profile is a 4x4 matrix M. It illustrates the situation, when the photos come from a new stage.

5) Apply convolution between P and M, as a result we will have a (N − 2s + 1)-length vector d, where the greater vector components indicate higher chance of a new stage (Fig. 7).

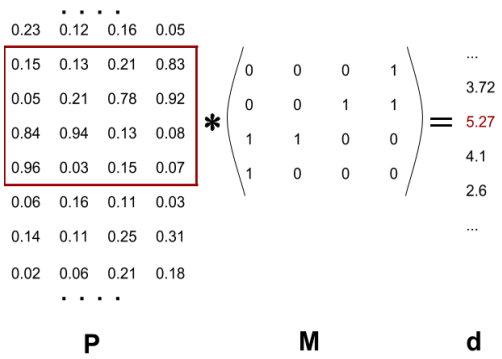

Figure 7. Progress detection scheme. Predictions matrix P is convolved with kernel M , resulting vector d is the difference measure.

6) Find all local maxima of the vector, each local maximum represents the progress update.

The method is motivated by the fact that a new incoming group of photos individually compared with previous photos should produce higher scores in the 1-Step-Net. This way, considering s × s pairs we expose more visual changes of the renovated environment. To evaluate the sequential stage classification method we sample consecutive photos from 6 bathrooms and 4 kitchens - 100 photos each. This gives us 6 and 4 progress ”ladders”; for every photo we can compute the stage disagreement error. The experiments showed that the window size 6 yields the best performance, having 59.6 % accuracy and 0.97 Kappa Index (Table 7). The 1-Step-Net models benefit from the sequential property of incoming photos and achieve a very high Kappa Index - 0.97 for the model with window size of 6.

Table 7. Performance with respect to the variable window size of the VGG19 and MobileNet based progress estimation models.

VGG19		Bathrooms		Kitchens	
	Window	Acc.	WKI	Acc	WKI
	2	45.5%	0.79	41.5%	0.61
	4	52.7%	0.93	42.1%	0.63
	6	**59.6%**	**0.97**	51.0%	**0.78**
	8	46.4%	0.93	**52.1%**	0.77
MobileNet					
	2	43.5%	0.76	40.5%	0.60
	4	51.2%	0.91	41.1%	0.65
	6	**58.9%**	**0.97**	**49.3%**	**0.81**
	8	56.8%	0.93	48.2%	0.80

Important to note that the window size should not be larger than the minimum number of photos in a new stage batch to avoid overlapping with a new stage. This way, the window size can serve as an adjustable parameter, depending on the size of the incoming batch of photos. The kitchen scenario demonstrates a lower performance compared to the bathroom case. It can be explained by the significantly lower number of training photos (3708 vs.1486). Nevertheless applying 1-Step-Net significantly improves the temporal precision of the estimation.

9. TIME COMPLEXITY

Table 8. Time complexity of the VGG19 and Mo- bileNet based progress estimation models.

	Comp. time for 100 photos (sec.)			
Window size	2	4	6	8
VGG19	7.12	14.35	21.86	27.92
MobileNet	1.65	3.2	4.71	6.33

The main computational burden comes from the processing of the incoming images by the deep neural network subunit. The processing time highly depends on the number and capacity of used GPU units. In our experiments we use two Tesla K80 graphics cards in parallel. For sequential stage identification in every iteration we process pairs of images. The number of unique pairs N p depends on the chosen window size s linearly (since we can reuse results of the previously computed pairs), and can be com- 2 puted with the formula $N_p = \frac{Ns}{4} - \frac{s^2}{4} - 1$, where N_p − number of pairs to process, N −the total number of photos, s− window size and $N >> s$. We analyze the time complexity of the stage identification for two models: VGG19 - the best performing model, and MobileNet - the lightest of studied models. Table 8

shows the processing time for 100 bathroom photos. One should take the values tentatively as computation speed can vary across graphic cards and software. We observe that MobileNet being a more compact network takes much less resources yielding a slightly worse performance overall, which makes it a more perspective candidate for applications.

10. DISCUSSION

Problem formulation: We created the Reno-2018 - an ordinal image classification dataset, designed to test work progress estimation methods. One alternative option was to define the task as a regression task: for every photo (or batch of photos) provide a % of completed work. The choice of defining the task as a classification, rather than a regression problem was made based on the following reasons. First, the image classification deep learning methods are more developed and understood. Second, photo acquisition was performed in irregular time slots, resulting in a more chronologically fragmented dataset. And third, the true completion degree holds ambiguity that is difficult to control for, namely should we take the actual time passed since the initial phase, or should we use an educated but still subjective progress estimation given by a construction manager. In total we collected ~ 7000 photos from 7 bathrooms and 6 kitchens that were originally distributed across 30 classes. Majority of the classes were underrepresented, having less than 50 photos. This led to the decision, to dismiss some classes and merge similar consecutive classes. We also kept the initial and the final stages, despite them being underrepresented, resulting in 10 classification stages and 3708 images for the bathrooms and 1486 images and 8 stages for the kitchens (see Table 1, 2).

Effect of the Ordering Net: One of the biggest challenges of the dataset is the non-strict nature of the ordering. Deep learning networks can learn that a certain set of features, most likely, defines a class. However, in our case, observable visual clues may not only be a result of the previous stage, but the previous stage itself might come from a different sequence. This fact possibly explains the low performance of the base CNN solutions. Deployment of the Ordering Net achieved both higher accuracy and Kappa Index. We assume that the better performance comes from the new complementary information brought by the Ordering Net. Indeed, the VGG19-CE net achieved a high top-2 accuracy without explicit use of the order information. Ensemble of two nets showed to improve Kappa Index when used with VGG19 − FC and both Kappa Index and accuracy when used with VGG19 − CE.

Effect of the 1-Step-Net: For the realistic progress estimation task (where test photos are given in a sequence) we trained a network similar to the Ordering Net, that specialized on dealing with consecutive stages - 1-Step-Net. With the information extracted by the network we can detect photo-time steps, when the process enters a new stage by filtering the sequence of network's outputs with an ideal kernel profile. This way, the sequence of recognized steps represents the current progress. The proposed method of sequential progress estimation significantly improved the temporal accuracy of the predictions. We believe that such an approach can be applied in various multistage scenarios to provide an image-based progress estimation, for example, it could be used tracking the mechanical assembly process.

Future work: We considered the scenario where smartphones are only used for data acquisition. Indeed, to the present day the most effective deep models require powerful GPU to run, but recent development of specialized libraries and GPU chips for mobile devices opens many opportunities for deployment of computer vision techniques on such devices. Since modern computer vision methods strongly rely on huge amount of samples, data collection techniques and strategies attract special interest. Potential work could include studying the transition of inference from static images to automatically recorded videos. Indeed, videos benefit from continuity of the recorded process but set the challenge of data storage and transferring.

11. CONCLUSIONS

In this paper, we propose a multistage deep learning network to estimate the progress of the renovation process based on smartphone photos taken by construction workers during the process. This system can track progress by detecting the changes in stages based on batches of collected photos. It can help analyze the dynamics of resource demand and save inspectors work hours by alarming in cases of crucial delays. According to evaluation with Reno-2018, a labeled photoset covering stages of bathroom and kitchen plumbing renovations, our solution achieves higher accuracy and better temporal agreement between the estimate and the true state compared to the baseline models. We believe that the proposed method can be applied to various progress estimation tasks.

12. ACKNOWLEDGMENTS

This work was a part of the reality capture for construction management project, funded by Business Finland. We would like to thank Xuebing Li, Andreas Hognabba, Markus Prittinen, Albert Lehtovirta and Otto Alhava for their help in creating the Reno-2018 dataset.

13. REFERENCES

[1] B. Albertina, M. Watson, C. Holback, R. Jarosz, S. Kirk, Y. Lee, and J. Lemmerman. Radiology data from the cancer genome atlas lung adenocarcinoma [tcga-luad] collection., 2016. *The Cancer Imaging Archive*.

[2] F. D. Anton, S. Anton, and T. Borangiu. *Integration of Visual Quality Control Services in Manufacturing Lines*, pages 329–342. Springer International Publishing, Cham, 2014.

[3] C. Beckham and C. Pal. A simple squared-error reformulation for ordinal classification. *arXiv e-prints*, abs/1612.00775, Dec. 2016.

[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-convolutional siamese networks for object tracking. In G. Hua and H. Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 850–865, Cham, 2016. Springer International Publishing.

[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems* 6, pages 737–744. Morgan-Kaufmann, 1994.

[6] M. BÃijgler, G. Ogunmakin, J. Teizer, P. A. Vela, and A. Borrmann. A comprehensive methodology for vision-based progress and activity estimation of excavation processes for productivity assessment. In *EG-ICE Workshop on Intelligent Computing in Engineering*, Cardiff, Wales, 2014.

[7] F. Chollet. Xception: Deep learning with depth-wise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[8] J. Cohen. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213–220, Oct. 1968.

[9] J. Cuadros and I. Sim. Eyepacs: An open source clinical communication system for eye care. In MedInfo, volume 107 of *Studies in Health Technology and Informatics,* pages 207–211. IOS Press, 2004

[10] H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6(6):1289–1301, 1994.

[11] H. Hamledari, B. McCabe, and S. Davari. Automated computer vision-based detection of components of under-construction indoor partitions. *Automation in Construction*, 74:78–94, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko,

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[15] L. Klein, N. Li, and B. Becerik-Gerber. Imaged-based verification of as-built documentation of operational buildings. *Automation in Construction*, 21:161–171, 2012.

[16] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In I*CML Deep Learning workshop*, 2015.

[17] M. Kopsida, I. Brilakis, and P. A. Vela. A review of automated construction progress monitoring and inspection methods. In *Proc. of the 32nd CIB W78 Conference 2015*, pages 421–431, 2015.

[18] J. Manyika, S. Ramaswamy, S. Khanna, H. Sarrazin, G. Pinkus, G. Sethupathy, and A. Yaffe. Digital America: a tale of the haves and have-mores. *Technical report, McKinsey & Company,* 2015.

[19] N. Mehdiyev, D. Enke, P. Fettke, and P. Loos. Evaluating forecasting methods by considering different accuracy measures. *Procedia Computer Science*, 95:264 – 271, 2016. Complex Adaptive Systems Los Angeles, CA November 2-4, 2016.

[20] I. Melekhov, J. Kannala, and E. Rahtu. Relative camera pose estimation using convolutional neural networks. *CoRR*, 2017.

[21] K. Ricanek and T. Tesafaye. Morph: a longitudinal image database of normal adult age-progression. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 341–345, April 2006.

[22] S. Roh, Z. Aziz, and F. Peña-Mora. An object-based 3d walk-through model for interior construction progress monitoring. *Automation in Construction*, 20(1):66–75, 2011.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

[24] D. Weimer, B. Scholz-Reiter, and M. Shpitalni. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals - Manufacturing Technology,* 65(1):417ÂŰ420, 2016.