DESIGNING SYNTHETIC OVERHEAD IMAGERY TO MATCH A TARGET GEOGRAPHIC REGION: PRELIMINARY RESULTS TRAINING DEEP LEARNING MODELS

Varun Nair[1],Paul Rhee[1], Bohao Huang[1], Kyle Bradbury[2], and Jordan M. Malof[1]

[1]Department of Electrical & Computer Engineering, Duke University, Durham, NC 27708
[2]Energy Initiative, Duke University, Durham, NC 27708

ABSTRACT

Convolutional Neural Networks (CNNs) have dominated performance on benchmark problems for object recognition in remote sensing imagery. However, recent work has shown that they may perform poorly when tested on imagery collected over a geographic location that was not present in its training imagery. In this work we explore the potential of designing synthetic overhead imagery to match a target geographic region (e.g., city or county), after which the synthetic imagery could be used to train deep learning models to recognize the unique visual features of objects/background in the target geographic location. We term this approach *geographic domain matching*. Towards this goal, we utilize a publicly-available dataset of synthetic overhead imagery, Synthinel-1. We systematically alter individual visual features of Synthinel-1 in an effort to match one real-world testing city: Vienna, Austria. We then evaluate whether these individual alterations improve the performance benefits of Synthinel-1 on Vienna and other cities. The results indicate that our proposed methodologies for altering the synthetic imagery seemed to lower its *overall* quality, but appear to improve the visual similarity between our synthetic imagery and the target city. These experiments are a first step towards developing methods for designing synthetic overhead imagery.

***Index Terms*—** overhead imagery, segmentation, domain adaptation, building segmentation

1. INTRODUCTION

Recently, Convolutional Neural Networks (CNNs) have dominated performance on benchmark problems for object recognition in remote sensing imagery [1]–[3]. However, most of these benchmark problems involve training and testing the models on imagery collected over roughly the same geographic locations – an approach we term *in-domain* testing. Recent studies [4]–[6] however have examined the performance of CNN models when they are tested on novel geographic locations that were not present in the training imagery – arguably a much more practical scenario that we term *cross-domain* testing. This work has indicated that performance is significantly worse in cross-domain testing, presenting an obstacle to more wide-spread application of recognition models to overhead imagery.

One frequently-cited reason for cross-domain performance loss is that many classes of objects (e.g., buildings, roads, vegetation) appear very differently in different geographic locations. In the computer vision literature these qualitative visual changes are often referred to as *visual domain shift* [7]. In this work we use the term *geographic domain shift*, to refer to the visual changes due to changes in the underlying scene content due to geographic changes.

1.1. Domain-matching synthetic imagery

One solution to address geographic domain shift is to collect a globally representative sample of overhead imagery, collected under varying lighting and weather conditions. However, this approach is infeasible due to the high costs of purchasing and hand-labeling (even small) quantities of satellite imagery. Recently, in [6], the authors developed Synthinel-1, a collection of synthetic overhead imagery encompassing several unique virtual world styles. Synthetic imagery refers to imagery that has been captured from a simulated camera operating over a virtual world. In a virtual world it is relatively easy to create large quantities of diverse imagery. The authors demonstrated that training with Synthinel-1 is beneficial when used to augment CNN training, especially for cross-domain testing. The authors conclude that the benefits are due to domain randomization [8], by presenting the model with more diverse imagery during training and increasing its overall robustness.

Fig. 1. Examples of synthetic imagery(a) and ground truth (b).

1.2. Contributions of this work

In this work we explore the potential of using synthetic imagery for *geographic domain matching*. In this strategy, we attempt to *design* synthetic imagery to mimic the visual features of a particular target domain (e.g., a city or county). While this approach requires design effort, we hypothesize that (if done well) it could yield much greater performance improvements than domain randomization, and would allow the training of much smaller models, on less data.

Towards this goal, we extract the imagery for a single style from Synthinel-1, which was called style "g" in [6]. We therefore term this baseline imagery as Synth1(g), which is illustrated in Fig. 1. We show that Synth1(g) improves *cross-*

domain performance on three real-world test cities, corroborating findings of [6] with a unique experiment. Subsequently, we systematically (i.e., only one at a time) alter individual visual features of Synth1(g) in an effort to match one real-world testing city: Vienna, Austria. We then evaluate whether these individual alterations improve the performance benefits of Synth1(g) specifically on Vienna, providing evidence that our proposed design methodologies were effective. Our goal is to take a first step towards developing methods for effectively designing synthetic overhead imagery.

2. EXPERIMENTAL METHODS

2.1. The Segmentation model

We use an encoder-decoder structure, with skip connections to maintain fine-grained object boundary details, as U-net in [9]. Inspired by [10] we use a larger ResNet-50 encoder. We train with a batch size of 7, following results with Synthinel-1 in [6]. The input size of the models is 512×512 pixels. For the Inria dataset, the input images are uniformly cropped into sub-images of the desired size as suggested by [11]. For all the models, we use the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and weight decay of $5e-4$ for 80 epochs. We use a learning rate of $1e-3$ and $1e-2$ for the encoders and decoders respectively and drop the learning rates by one magnitude after 50 epochs. We use a smaller learning rate for encoders since they are already pre-trained.

2.2. Publicly-available building segmentation datasets

Inria [5]. The INRIA Aerial Image Labeling Challenge Dataset is a recent benchmark dataset for building segmentation. It features $0.3m^2$ resolution RGB aerial imagery collected over ten cities across the U.S. and Europe. A total of 81 km^2 of imagery is available for each city. We conduct our experiments on a subset of five cities for which publicly available ground truth labels are available: Austin, Chicago, Kitsap, Western Tyrol, and Vienna.

Synthinel-1[6] is composed of synthetic color overhead imagery collected at 0.3m resolution over nine virtual worlds with different styles. Synthinel-1 was created is publicly-available, along with Python code to extract overhead imagery from a given virtual world. For our experiments we used approximately $8km^2$ of imagery from just one city style: "Austin city style".

2.3. The MRS framework for deep learning

To conduct our experiments we use a publicly-available[1] PyTorch framework called "MRS" that has many functions specifically designed for deep learning tasks on remote sensing data.

3. ALTERNING SYNTHETIC IMAGERY TO MATCH A TARGET

In this section we describe our methodology for altering the virtual worlds to match our target city: Vienna, Austria. All of our virtual worlds were generated using the CityEngine software package[2], and following guidance and publicly-available software from [6]. The alterations here are described qualitatively due to space limitations, but we will release our code with this publication to ensure repeatability.

Road Network Alterations. This change involved the redesign of the road network topology of the virtual world to match Vienna. The Synth1(g) road network included a mix of street patterns built into CityEngine: hexagonal, raster, organic, and radial. This is illustrated in Fig. 2 (left). These patterns did not match well (qualitatively) with those present in Vienna and we found that using Organic major street pattern and Organic minor street pattern were qualitatively much more similar. This updated street pattern is shown in Fig. 2 (right).

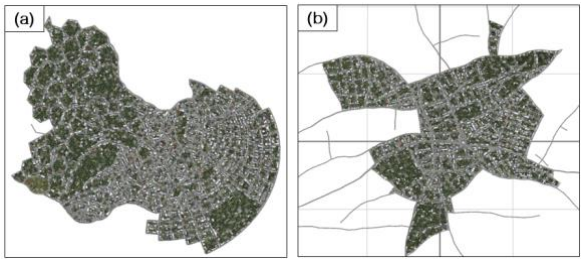

Fig. 2. Illustration of road network alterations. (a) original road network, and (b) altered road network.

Building Shape Alterations. The rule file used to generate this version of the city was changed to better reflect the distribution of different shapes of buildings in the real Austin. We incorporated O-shaped buildings, which were observed in much of Vienna's urban scenes

Light Alterations. The light intensity, angle (height from horizon), and direction were changed to visually match those of Vienna. We found that we only needed to vary the light angle from 60 degrees (the default) to 50 degrees, making the shadows somewhat longer.

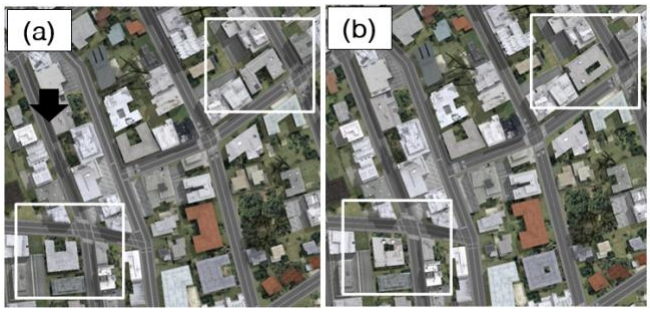

Fig. 3. Illustration of building shape alterations. (a) original "u-shape" buildings (white boxes), and (b) altered "o-shape" buildings.

Rooftop texture and color alterations. To create the Synth1(g) virtual world, the authors in [6] used a set of textures that were available already in CityEngine; this bank was not chosen to match Vienna. Here we extracted five patches of rooftop imagery from real-world imagery over Vienna, and then added these patches to the bank of

[1] https://github.com/bohaohuang/mrs

[2] https://www.esri.com/en-us/arcgis/products/esri-cityengine/overview

textures/colors already used to create Synth1(g). In practice we believe that it should be cost-effective and fast to extract a few patches of real-world rooftop imagery, rather than purchasing large quantities of real-world imagery and annotating all the rooftops. We hypothesize that we may be able to generate many real-world variations of the textures in the real-world imagery using a few samples taken from a real-world location.

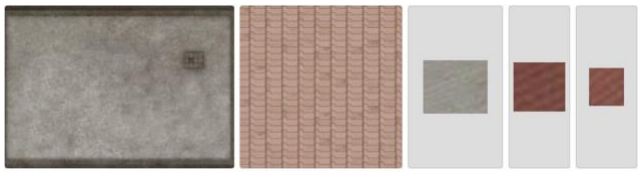

Fig. 4. The bank of five textures/colors extracted from rooftops in the real-world Vienna imagery.

Our approach here is a simple initial methodology, by which we extract these patches and add them to the bank of textures used to create Synth1(g); these textures/colors are extracted pseudo-randomly from the total available bank and mapped onto the 3-dimensional shapes of the buildings in the virtual world. Fig. 4 illustrates the bank of real-world textures we extracted from Vienna, and Fig. 5 shows examples in which the new textures were sampled for rooftops.

Fig. 5. Illustration of rooftop texture/color alterations. (a) Original textures of buildings (white boxes), and (b) the same rooftops with textures extracted from real-world imagery.

4. EXPERIMENTAL GOALS AND DESIGN

In this work we will use Intersection over Union (IoU) as the performance metric. It measures the similarity between the predicted region and the truth region by $\text{IoU} = \frac{A \cap B}{A \cup B}$ where A is the predicted region and B is the truth region. We design our experiments to answer the following scientific questions: (i) How well does Synth1(g) perform, and does it improve over cross-domain testing using only real-world imagery? (ii) Can we alter the synthetic imagery to match the visual features of a particular city in a novel geographic domain? (iii) What is the best we could perform in the target city, if we designed the synthetic city perfectly?

Given these objectives, our experimental design is presented below in Fig. 7. We split the five available cities in the Inria benchmark dataset into two sets; {Kitsap, Tyrol-W} are used for training, and {Austin, Vienna, Chicago} are used for testing. This design emulates a cross-domain testing scenario, in which we test on three cities (i.e., geographic domains) that are not present in the training imagery. During training we used mixed-batch training, following [6], in which 1 of the 7 images used in each training mini-batch are sourced from a pool of synthetic imagery. However, in our experiments we will vary the source of this single patch, as illustrated in the 2nd column of Fig. 7.

Experiment name	Variable in training (1/7 of each mini-batch)	Fixed in training (6/7 of each mini-batch)	Testing data
Control-real	Kitsap, Tyrol-W	Kitsap, Tyrol-W	Austin, Vienna, Chicago
Control-Synthetic	Synthetic generic	Kitsap, Tyrol-W	Austin, Vienna, Chicago
Control-Oracle (X)	**Real** target city (X)	Kitsap, Tyrol-W	Austin, Vienna, Chicago
Synthetic (X,Y)	**Synthetic** target city (X,Y)	Kitsap, Tyrol-W	Austin, Vienna, Chicago

Fig. 6. Experimental design

In order to answer question (ii), we use the Synthetic(X,Y) experiment design, in which we generate synthetic imagery with alterations of type Y, which are designed to match target city X. The alterations we consider are described in Section 3; for example, Y=Light indicates that we changed the lighting of Synth1(g) to match the target city X (in our case X=Vienna). The Control-Oracle experiment is designed to answer question (iii) and sources the single mixed-batch image from the *target* domain, X, providing an estimate of the best-case scenario, in which our synthetic imagery perfectly matches X.

5. EXPERIMENTAL RESULTS

5.1. Control results

Table 1 presents the performance, in terms of IoU for the three control experiments. Based on the results we see that the Synth1(g) imagery provides improved IoU on all three of the unseen testing cities, consistent with [6]. Here we used a different experimental design and deep learning model than in [6], thereby providing additional evidence for the conclusions from [6] that the Synthinel-1 imagery (or, in this case, the Synth1(g)) is beneficial for cross-domain testing. The Control-Oracle experiment indicates that replacing Synth1(g) with real-world imagery from the target domain substantially improves performance on the target domain city (Vienna), as expected. It also improves performance on the other testing cities, to a similar degree as the Synthetic-Control.

Table 1: Results for "Control" experiments.

Experiment Name	Test City IoU		
	Austin	Chicago	Vienna
Control-Real	64.35	58.13	72.02
Control-Synthetic	65.36	59.81	73.13
Control-Oracle(Vienna)	64.97	60.90	81.62

5.2. Domain matching to Vienna

Table 2 presents the results in which we attempted to alter the Synth1(g) synthetic imagery to match the domain of Vienna. The results indicate that two of the four individual alterations were beneficial, however, the IoU improvements were modest, and it is unclear whether they are within the margin of error (i.e., explained away by other sources of performance variation). We also know from the Control-Oracle experiment that this limited performance improvement is not simply due to the difficulty of Vienna, because we achieve substantially greater performance when training (in the exact same mixed-batch scheme) using real-world imagery from Vienna.

It is notable however that our changes consistently reduce performance substantially on the other non-target testing cities. This seems to suggest that our methodology for altering Synth1(g) is tending to reduce the general quality of Synth1(g), because IoU tends to reduce substantially when we alter it. However, despite this general reduction in quality, we still perform similarly, or better, on the target city, suggesting that we may be matching (to some degree) the target domain, mitigating the general detrimental effects of our changes.

Also, it is notable that changing the building shapes and textures, respectively, seems to have the greatest impact on IoU (both negative and positive), suggesting that these visual features are (relatively) important in virtual world design.

Table 2: Results for the Variable experiments.

Experiment Name ($C_{tgt} = Vienna$)	Test City IoU		
	Austin	Chicago	Vienna
Control-Synthetic	65.36	59.81	73.13
Synthetic(Vienna,Road)	60.32	52.40	72.58
Synthetic(Vienna,Shape)	57.00	52.33	**73.30**
Synthetic(Vienna,Light)	61.38	53.86	72.81
Synthetic(Vienna,Textures)	57.53	53.52	**73.29**
Synthetic(Vienna,All)	61.59	56.34	**73.72**

6. CONCLUSIONS

In this work we attempted to *design* synthetic overhead imagery to mimic the visual features of a particular target domain (e.g., a city or county), in order to improve performance of deep learning models on that target domain. We systematically altered a publicly-available dataset Synthinel-1 [6] to match Vienna, Austria. The results indicated that our proposed methodologies for altering the synthetic imagery seemed to lower its *overall* quality, but appear to improve the visual similarity between our synthetic imagery and the target city. Overall our changes resulted in a mix of modest improvements and losses in performance on Vienna. These experiments represent a first step towards developing methods for designing synthetic overhead imagery, and understanding which factors are most important in doing so.

7. ACKNOWLEDGEMENTS

We want to thank the NVIDIA corporation for donating the graphics processing unit (GPU) for this work. Bohao Huang would like to thank the Energy Data Analytics Ph.D. Fellowship program from the Duke University Energy Initiative funded by the Alfred P. Sloan Foundation for supporting his work.

8. REFERENCES

[1] I. Demir *et al.*, "DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images," pp. 172–181, 2018.

[2] V. Iglovikov, S. Mushinskiy, and V. Osin, "Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition," vol. June, 2017.

[3] B. Huang *et al.*, "Large-scale semantic classification: outcome of the first year of inria aerial image labeling benchmark," in *International Geoscience and Remote Sensing Symposium*, 2018.

[4] R. Wang, J. Camilo, L. M. Collins, K. Bradbury, and J. M. Malof, "The poor generalization of deep convolutional networks to aerial imagery from new geographic locations: an empirical study with solar array detection," *2017 IEEE Appl. Imag. Pattern Recognit. Work.*, pp. 1–8, 2018.

[5] E. Maggiori *et al.*, "Can Semantic Labeling Methods Generalize to Any City ? The Inria Aerial Image Labeling Benchmark To cite this version :," pp. 3226–3229, 2017.

[6] F. Kong, B. Huang, K. Bradbury, and J. M. Malof, "The Synthinel-1 dataset : a collection of high resolution synthetic overhead imagery for building segmentation," pp. 1–10.

[7] B. Sun and K. Saenoko, "Deep CORAL: Correlation Alignment for Deep Domain Adaptation," *Comput. Vis. – ECCV 2016 Work.*, vol. 9915, pp. 738–752, 2016.

[8] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2017-Septe, pp. 23–30, 2017.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Miccai*, pp. 234–241, 2015.

[10] V. I. Iglovikov, S. Seferbekov, A. V. Buslaev, and A. Shvets, "TernausNetV2: Fully Convolutional Network for Instance Segmentation," 2018.

[11] B. Huang, D. Reichman, L. M. Collins, K. Bradbury, and J. M. Malof, "Sampling training images from a uniform grid improves the performance and learning speed of deep convolutional segmentation networks on large aerial imagery," in *IGARSS*, 2018.