Visual Classification of Blighted Properties

Eduardo Yap
University of Minnesota
Minneapolis, MN
yap00014@umn.edu

Zaffer Hussein
University of Minnesota
Minneapolis, MN
husse147@umn.edu

Abstract

In the context of housing, blight can be described as a state of deterioration or attractive nuisance. Detection of blighted areas is important for slowing the spread of urban decay and property devaluation across a city. To combat the spread of blight, we propose a computer vision solution. First, we present a novel dataset that labels property images into blighted or non-blighted. The task of classifying these images falls into fine-grained object classification. There are nuanced visual differences within the two classes that make it difficult for standard object classification models to achieve an outstanding accuracy. In this paper, we propose the use of visual attention mechanisms to address this issue. We first explore an existing attention model to increase global attentiveness to distinctive features. We then implement a canvas generation module to apply the previous attention model to different image regions.

1. Introduction

Property blight is currently a big problem for many major urban cities and can have costly effects. As houses deteriorate to the point of no return, the overall value of the land declines, causing subsequent deterioration and abandonment of nearby properties. Blighted properties can also lead to increased neighborhood crime, higher risk of accidental fires, and public and health nuisances [1]. As of 2014, more than a fifth's of Detroit's properties were affected by blight and over 40,000 houses were ordered for demolition [3]

Figure 1. Blighted houses in a Detroit neighborhood.

As the level of blight becomes unmanageable, big cities such as Detroit and New Orleans have resorted to data-driven approaches to combat blight in different ways. These solutions have seen a reasonable amount of success, but have their downsides. For example, applications have been created for users to send their own pictures of houses along with their own assessment of the property condition [28, 4]. These initiatives have helped map out the blight distribution across a city. However, one problem with this approach is that evaluations can be inconsistent across users and biases may be present[20].

Machine learning approaches have been considered as well. One approach used historical and geographical data of home inspections in Cincinnati, OH to predict future blight violations [7]. Another approach utilized Detroit blight ticket history, which included geographical and economical data, to predict future blight violations [21]. While these approaches can shed some light on factors that lead to blight, they can lead to unintentional discrimination to certain groups of people. It's also not viable for cities with no previous/limited blight property records.

In this paper, we propose a computer vision solution which classifies between blighted and non-blighted properties. There are multiple challenges we encounter in this paper. First, there are no existing publicly available datasets that fit our problem description, so we have to create our own dataset. Second, we run into the problem of classifying images that are visually and semantically similar. Blighted features do not encompass the whole property so classification methods must be able to pick on the small regions of blighted features within an image. This is known as fine-grained object classification.

In this paper, we outline our process of building a comprehensive dataset that can be used to train models aiming to identify blight. We also explore the performance of different models in our dataset. We evaluate the performance on a baseline VGG model, an existing attention model, and a modified version of the attention model using a canvas generation module.

2. Related work

Building Recognition. Computer vision techniques have been previously used for tasks of building classification, regression and segmentation. The first applications of computer vision to building imagery simply focused on building recognition [34, 15, 27, 16]. Motivated by using landmarks for localization and navigation, Zhang and Kosecka introduced a localized color histogram representation to build a hierarchical scheme comprised of two stages, which produced better results than previous work [34]. Alternatively, Li and Allinson used local orientated features to apply a steerable filtered-based building recognition (SFBR) model to this problem [15]. From an aerial approach, Thiele et al. worked to detect line structures of building edges in interferometric synthetic-aperture (SAR) data [27]. Building reconstruction was achieved by fusing these line structures and edges to assemble polygons representing building structures, resulting in varying qualities of recognition, which was less successful for smaller buildings and more complex buildings [27]. Furthermore, Li and Shapiro developed and applied consistent-line-cluster features to building recognition [16]. In this approach, Edges were segmented and clustered into spatial groups based on orientation, color and spatial features [16].

Building Classification and Regression. After the rise of neural network architectures such as AlexNet [13], VGG16 [26] and ResNet [6], fine-tuning object recognition/detection tasks to use the power of CNNs has become easier than ever. Multiple papers have been recently written using these architectures with a focus on building classification/regression. Koch et al. extract patches out of building images and feed them to ResNet in order to estimate the age of the building [33]. They also use the same technique to predict a building's structural condition in order to assess its real estate valuation [11]. Poursaeed et al. utilize various real estate images from interior to exterior house pictures to predict the actual price of a property [14]. These papers share the same approach of extracting high-level features of out the house in order to make fine-grained predictions.

Post-hoc Attention in CNNs. There are two ways attention mechanisms can be used in a CNN. First, attention mechanisms can be used as a post-hoc analysis of the model. Attention maps are used as a method for visualizing and interpreting convolutional neural networks. They're defined as "a scalar matrix representing the relative importance of layer activations at different 2D spatial locations with respect to the target task" [8]. One example of a post-hoc attention analysis technique is Grad-CAM, which focuses on capturing the gradient information flowing into the last output layer of a network to produce a localization map of the most important regions in the image [24]. The second way attention mechanisms can be used is explained in the next subsection in the context of fine-grained classification.

Attention for Fine-Grained Classification. One of the main challenges of performing classification on building images is the small visual differences between classes. Classifying images with subtle differences between labels is known as fine-grained classification. Standard models such as VGG16 and ResNet suffer difficulties trying to discriminate between fine-grained images, because features from the images are extracted in a global fashion. To combat this, trainable attention models that selectively focus on the objects of interest have been proposed. Seo et al. propose an attention model of this kind, known as Progressive Attention Networks (PAN). Here, attention mechanisms are introduced to various stages of a Convolutional Neural Network to enable "precise attention over objects of different scales and shapes" [25]. Jetley et al. introduce attention submodules to VGG16 and ResNet and see performance gains over the base models [8]. Zhao et al. propose Diversified Visual Attention Networks (DVAN) that use LTSMs to learn the attentiveness and discrimination of diverse image canvases [35]. Yang et al. propose a multi-agent cooperation approach, where the agents work together to localize informative regions in the image [32]. Wang et al. use Residual Attention Networks (RAN), a model built by stacking multiple Attention Modules together [30].

The use of attention has seen success in other domains such as pancreas segmentation [19], ultrasound classification [23], and movie story question answering [10]. They've also seen great success in popular fine-grained classification datasets, such as Caltech-UCSD Birds 200 [31], Stanford Dogs [9], and Stanford Cars [12]. We expect to see the same success with the blight classification task.

3. Data

Despite the previous work done on housing images, there weren't any openly available datasets that fit our problem description. We created our own dataset. The dataset consists of images containing properties. Properties are surrounded by bounding boxes labeled blighted or non-blighted. The final dataset consists of 895 blighted properties and 791 non-blighted properties.

3.1. Address Collection

The first step was to find addresses of properties that had been flagged as blighted by a governmental body. The cities of Richmond, IN; Baton Rouge, LA; Des Moines, IA; Detroit, MI; and Houston, TX all published online city maps that geolocated properties that had received blight violations [5, 17, 2, 4, 18]. From these maps, we were able to extract address and violation date information for hundreds of properties. Some maps provided their own property images, but for most of the addresses, we had to hunt for the images ourselves. We also found a Kaggle dataset containing over 300 000+ rows, where each row corresponded to

a single blight ticket. Each row had date and address information that we added to our collection of addresses.

Figure 2. Top: blighted houses. Bottom: non-blighted houses.

3.2. Image Collection

To collect images, we used Google Maps Street View to visit the addresses collected. Thanks to the time machine feature, we can choose the location date closest to the violation date. For example, if a blight ticket was issued on September 2013, we can look at the street view of the address from say, August 2013. We captured multiple houses using different angles and by taking a full screen shot of the entire Google Maps window. If possible, we also took pictures of the same house at different times in order to capture different surroundings, lighting, camera resolutions and stages of house deterioration. To capture non-blighted houses, we took pictures of houses from the same neighborhoods as those that were blighted.

3.3. Image Annotation

Since we took large screenshots of the Street View locations, we had to draw bounding boxes around each house and label them. To do this, we used the tool labelImg to label and draw bounding boxes around each house, and save them in YOLO format [29]. Though we don't use the bounding boxes during training of our models, we still decided to enrich the dataset with bounding boxes for future use. The dataset could be used in the future for house detection tasks. To finalize the data collection process, we wrote a script to crop the images using the bounding box values to obtain our final complete dataset.

3.4. Data Limitations

One of the possible limitations from our approach to collecting data is the different standards cities use to categorize a property as blighted. What some cities may deem as a blighted property, others may see it as a non-blighted property due to different working definitions. This challenge blurs the line even further between the two categories and it might be impossible for a neural network to compensate for this during training. In the future, models should be trained on data solely from one city and one standard set of definitions. Unfortunately, there wasn't enough data out there for us to focus in one city alone.

4. Methods

4.1. Baseline Method

As a baseline, we use VGG16 as our based model with weights pre-trained on ImageNet [13, 26]. The layers were then frozen, so that the weights didn't change while training. The top fully connected layers were replaced by a new fully connected head with a softmax output of size 2 for our 2 labels (blighted/non-blighted).

4.2. Approach Overview

To tackle the fine-grained classification task in our dataset, we propose the use of an attention model, that is a model that selectively focuses on discriminative regions of an image through the use of some attention module. This approach is based on the assumption that it's helpful to identify salient image regions and amplify their influence on the final output of the model. For our paper, we use the model presented in the paper "Learn to Pay Attention" by Zentley et al. [8]. The paper introduces some modifications to the existing VGG16 network. It introduces attention submodules at different convolutional layers that exploit the image information used by the CNN to make a decision. These will be explained further in section 4.3.

We also propose a further modification to the model above. Inspired by Zhao et al.'s paper, multiple canvases are extracted from the input that are then fed to separate attention models [35]. The aim of this is to capture more fine-grained discriminative features in different canvases that would then be used to calculate a final prediction. This modification is explained in section 4.4.

4.3. Learn to Pay Attention

The "Learn to Pay Attention" paper introduces attention submodules to a standard VGG16 architecture. Their model was able to outperform other attention models in the fine-grained CUB-200-2011 dataset. Although multiple modifications are introduced and tested in the paper, we only explore one of the variations that worked best for our dataset.

Consider the VGG16 architecture, as shown in Figure 3.a. First, two additional convolutional layers (512) are added after the final max-pooling layer. Then, the first two max-pooling layers are moved in front of these two newly added convolutional layers. The reasoning behind this is to keep the initial layers used for estimating attention at a higher resolution. The dense layers are replaced by a single dense layer of output 512.

Now, we'll explain the paper's design of the attention submodule. The output of the final dense layer is denoted as g. This is the global feature vector. Let $\mathcal{L}^s = \{\ell_1^s, \ell_2^s, ..., \ell_n^s\}$ denote the set of feature vectors extracted at each max-pooling layer $s \in \{1, ..., S\}$, where n is the total spatial locations of the input. The set of compatibility

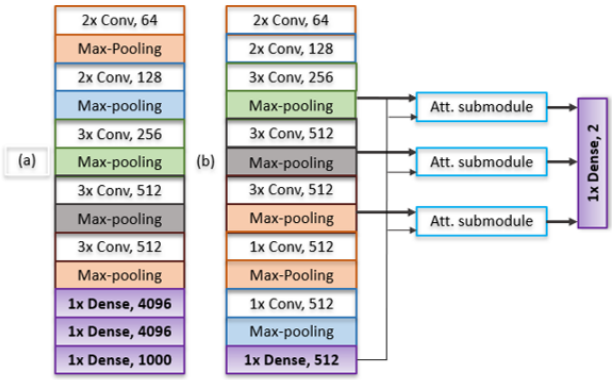

Figure 3. (a) Standard VGG architecture. (b) att-VGG architecture.

scores can then be defined as $C(\hat{\mathcal{L}}^s, g) = \{c_1^s, c_2^s, ..., c_n^s\}$, where

$$c_i^s =< \mathbf{u}, \ell_i^s + \mathbf{g} >$$

The compatibility score is then the result of a fully connected mapping from the addition of g to each of the feature vectors in s. $\mathbf{u}$ is the set of weights of this fully connected mapping. $\hat{\mathcal{L}}^s$ is an image of a mapping from $\mathcal{L}^s$ to a set of vectors that are of the dimensionality of $\mathbf{g}$. In our case, only the output of the first max-pooling layer (256) has to be mapped to the dimensionality of g (512). This is done through a trainable convolutional layer of output 512. The other 2 max-pooling layers are already in the dimensionality of $\mathbf{g}$. Then the set of compatibility scores are normalized by a softmax layer:

$$a_i^s = \frac{\exp\left(c_i^s\right)}{\sum_{j=1}^{n} \exp c_j^s}, i = \{1, ..., n\}$$

These normalized scores are then used to produce a single vector

$$g_a^s = \sum_{i=1}^{n} a_i^s * l_i^s$$

Finally, the set of vectors $g_a = \{g_a^1, ..., g_a^S\}$ are concatenated and fed to a final dense layer that outputs the classification prediction. In our case, $S = 3$, meaning that we introduce 3 attention submodules at 3 different convolutional layers. The final dense layer has an output vector of size 2 representing our 2 classes, blight and non-blighted. We'll refer to this architecture as att-VGG from now on.

4.4. Canvas Generation

While attention is implemented through the architecture defined in section 4.3, the inputs to the attention submodules are taken from global image vectors. To diversify and utilize attention to its fullest extent, we take an approach similar to the one outlined in Zhao et al's paper [35]. Through this approach, multiple canvases of different window sizes and locations are taken from the input image and fed to separate att-VGG models, enhancing the contributions of more fine-grained details. An overview of the approach can be seen in Figure 4. We will refer to this model as Canvas att-VGG from now on.

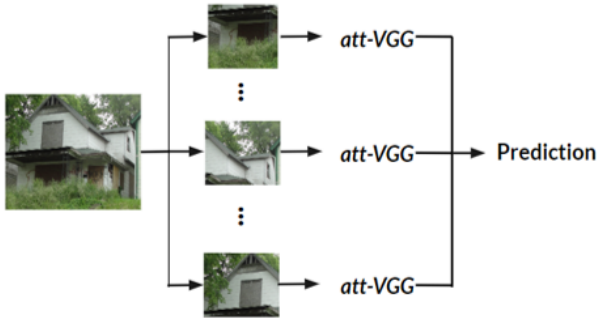

Figure 4. Canvas att-VGG architecture.

Window sizes are defined with corresponding strides. Multiple window-sized canvases are cropped from the input image. The number of canvases to take is determined by the stride. For example, let's say we have an input image of size 256x256. Setting the window size to 224 and the stride to 32 will generate 4 different canvases of size 224x224. To increase diversity, we obtain multiple canvases at different window sizes. All canvases are ultimately resized to a set resolution before being fed to the corresponding att-VGG.

The separate att-VGG are all trained from scratch on different locations of the input training images. Each att-VGG will output a prediction vector. There are two approaches that can be taken to compute the final prediction vector. First, we can simply add them all element-wise. Or, we can determine the label prediction of each vector and do majority-voting to determine the label prediction of the overall model. These two approaches are explored.

The intuition behind the proposed canvas generation module is that there is no one single feature that differentiates blighted and non-blighted properties. Broken windows, unkempt lawns, deteriorated walls, missing walls, etc. are all visual features that can be present or not. To attend to the variety of features, we train canvases that focus on different house regions.

5. Experimental Setup

In this section, we evaluate the performance of the baseline, att-VGG and canvas att-VGG models on the blighted property dataset. We'll then discuss the results.

5.1. Data

As mentioned earlier, the dataset consists of 918 images labeled as non-blighted and 918 images labeled as blighted. The dataset is divided into train, validation and test sets using a 60/20/20 split. Table 1 shows the exact split.

	Train	Validation	Test
Blighted	542	195	187
Non-blighted	559	172	181

Table 1. Train-validation split.

5.2. Implementation Details

For the implementation of the models, we assume an input image resolution of 120x120. For pre-processing, images are first re-sized to this resolution and normalized by the mean of their RGB values. Resizing is done using nearest neighbor interpolation. All training is done on an NVIDIA Tesla T4.

The baseline VGG model is implemented using Keras. It's initialised with "imagenet" weights. The model is trained for 50 epochs using stochastic gradient descent with a momentum of 0.9 and learning rate of 0.0001. We use a batch size of 32.

For att-VGG, we use SaoYan's Pytorch implementation found on GitHub [22]. The model is trained from scratch and the weights are initialised using Xavier initialisation. The model is trained for 50 epochs using stochastic gradient descent with a momentum of 0.9, learning rate of 0.0001 and a $5e^{-4}$ weight decay. We use a batch size of 8.

For canvas att-VGG, we use three different window sizes: 224x224, 168x168, and 112x112. The strides are set to 32, 44, and 48, respectively. In total, we get 29 canvases and therefore, 29 att-VGG models that have to be trained. For pre-processing, the input image is first re-sized to 256x256. The canvases are extracted, then resized to 120x120, and fed to the respective model. The training parameters remain the same.

For calculating the final prediction label, we evaluate two methods. The first method is majority-voting where we tally up the predictions and the label with the majority of the votes gets chosen as the final prediction label. The second method is to add up the dense output of all the models and see which prediction label has the highest aggregation.

5.3. Quantitative Results

Table 2 shows the performance on the test set for all the models. Recall that the test set consists of 187 blighted images and 181 non-blighted images. att-VGG (75.54%) sees a great improvement from the baseline VGG (70.11%). By training att-VGG on multiple canvases we see another great improvement: 79.89% for majority voting method and 80.43% for aggregation method.

Model	Accuracy
VGG	70.11
att-VGG	75.54
Canvas att-VGG (majority voting)	79.89
Canvas att-VGG (aggregation)	80.43

Table 2. Model Evaluation on Dataset

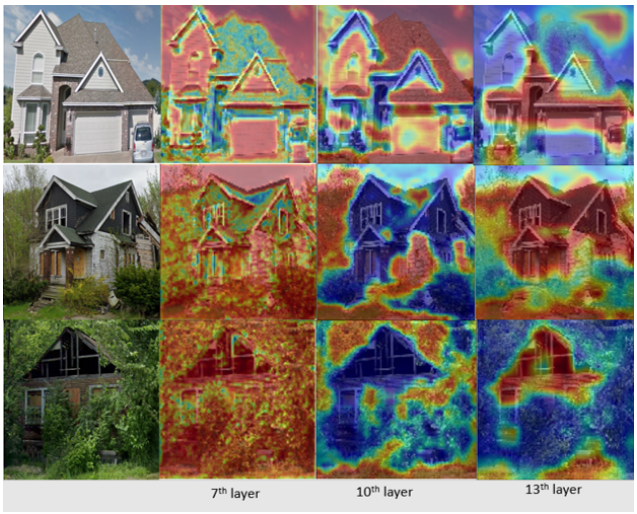

Figure 5. Layer activation maps.

5.4. Qualitative Results

For att-VGG, we visualize the attention maps for the convolutional layers at which the attention submodules were introduced on. These are the 7th, 10th and 13th layers. Specifically, these are the set of compatibility scores $C(\hat{\mathcal{L}}^s, \mathbf{g})$ that were learned during the training process. Figure 5 shows the attention maps for some examples.

The seventh layer seems to be focusing on some select spots. It's difficult to interpret why the model chooses to focus on these seemingly random areas. The tenth layer seems to be focusing on the background of the house. It seems that elements surrounding the house such as pristine lawns, tall grass, and street cleanliness can help the model determine the blight-ness of the property itself. The 13th layer seems to be focusing its attention on the actual house and intrinsic elements that might show "deterioration" or "cleanliness". For example, for the third example, the model focuses its attention on the missing walls on the roof wall.

Although the visualizations are somewhat interpretable, they're also seemingly random for some activation areas. We believe that this is because there are multiple distinctive features that can differentiate the two classes. This causes the attention modules to be unable to attend to one single feature and instead has to focus on more general features such as the surrounding background or the entire house. Of course, the Canvas att-VGG model tries to address this problem, but this could potentially be solved by introducing

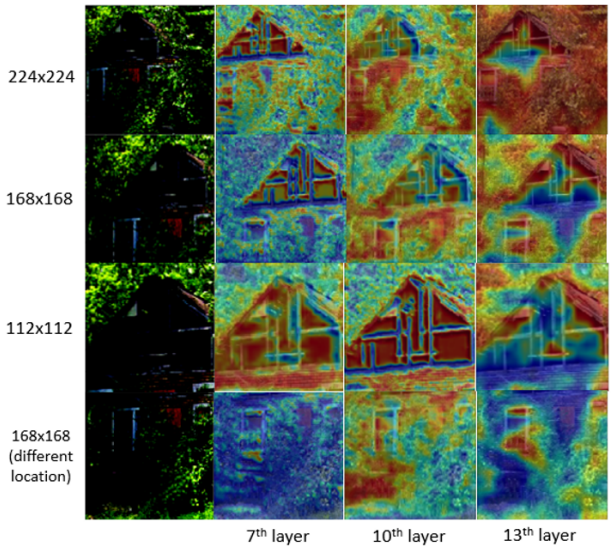

Figure 6. Layer activation maps at different image window sizes.

more attention modules at more convolutional layers in att-VGG. By splitting the tasks between more attention modules, we can expect att-VGG to perform better, but this is just speculation for now.

The same visualizations can be extracted from Canvas att-VGG for each of the models focused on different image canvases. Figure 6 shows an example input image on 3 canvases at different window sizes. It seems that the seventh layer now focuses on the wall and roof of the property, when it unclear what it was attending to in att-VGG. Possibly, the higher scale has allowed the layer to better focus. The tenth layer seems to be the one focusing on the overall property itself, while the 13th layer focuses on more background objects. As the scale increases from 224x224 to 112x112, it seems that the layer is able to focus on more distinct features and defined features. The last row shows one of the 168x168 canvases that's focused on a different location than the examples above it. The seventh layer focuses on the small rough edges on the door this time. We see that by having different canvases, the models can gather attention from different image locations and let them contribute to the overall prediction.

Of course, these attention maps are not perfect and could benefit from different approaches. More window sizes can be added to aid in the recognition of fine-grained details. Additionally, instead of manually extracting canvases at set window sizes and strides, we could engineer keypoint extractors that can guide us in selecting relevant regions at appropriate scales, while disposing of those that are a hindrance to the model.

6. Conclusion

We propose a computer vision solution to the classification of blighted properties. Through the use of online city maps, we built a comprehensive dataset of over 1800 properties. To address the fine-grained classification task of the dataset, we propose an end-to-end trainable attention model that utilizes Canvas generation to enhance the performance of an existing attention model. The Canvas att-VGG seems to attend better to the existing house features than att-VGG. This is evident by the better accuracy results and more well-defined visualizations.

For the future, we can experiment with many approaches to increase the accuracy. As mentioned before, we can collect even more data or limit the training of the model to a single city in order to eliminate potential bias. We could also add more canvases, or find a more meticulous way to generate optimal canvases rather than manually extracting them.

Overall, we believe that our solution could be used by city governments in the fight against blight. It addresses the problems present in existing solutions by avoiding the manual assessment of a property, eliminating some potential human bias, and steering clear of potential discrimination against certain groups of people.

7. Contributions

Zaffer contributed to the mass collection and annotation of property images as well as implementing image preprocessing and visualization functions. Eduardo also contributed to the collection and annotation of images. Eduardo also contributed in the implementation of the baseline VGG model and the canvas generation architecture. Eduardo also refactored SaoYan's "Learn to Pay Attention" Pytorch implementation to fit our paper's problem description [22].

References

[1] National Vacant Properties Campaign. The national vacant properties campaign. vacant properties: The true costs to communities, 2005. 1

[2] IA City of Des Moines. Blitz on blight. https://gisweb3.co.wayne.in.us/BEP/. 2

[3] Monica Davey. Detroit urged to tear down 40,000 buildings, 2004. 1

[4] Detroit Blight Elimination Task Force. Motor city mapping. https://motorcitymapping.org. 1, 2

[5] Wayne County Indiana Government. Richmond indiana blight elimination program. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[7] Bhavika Reddy Jalli, Adam Rauh, Xinyu Tan, Jared Webb, Joshua Bochu, Arya Farahi, Danai Koutra, Jonathan Stroud,

and Colin Tan. Understanding blight ticket compliance in detroit. 1

[8] Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip HS Torr. Learn to pay attention. *arXiv preprint arXiv:1804.02391*, 2018. 2, 3

[9] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011. 2

[10] Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D Yoo. Progressive attention memory network for movie story question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8337–8346, 2019. 2

[11] David Koch, Miroslav Despotovic, Muntaha Sakeena, Mario Döller, and Matthias Zeppelzauer. Visual estimation of building condition with patch-level convnets. In *Proceedings of the 2018 ACM Workshop on Multimedia for Real Estate Tech*, pages 12–17, 2018. 2

[12] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 2

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2, 3

[14] Stefan Lee, Nicolas Maisonneuve, David Crandall, Alexei A Efros, and Josef Sivic. Linking past to present: Discovering style in two centuries of architecture. 2015. 2

[15] Jing Li and Nigel Allinson. Building recognition using local oriented features. *IEEE Transactions on Industrial Informatics*, 9(3):1697–1704, 2013. 2

[16] Yi Li and Linda G Shapiro. Consistent line clusters for building recognition in cbir. In *Object recognition supported by user interaction for service robots*, volume 3, pages 952–956. IEEE, 2002. 2

[17] City of Baton Rouge. Blighted property dashboard. https://gismaps.brla.gov/. 2

[18] City of Houston. 311 blight map. http://www.houstontx.gov/fighthoustonblight/blightmap.html. 2

[19] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018. 2

[20] Bradley Pough and Qian Wan. Data analytics and the fight against housing blight: A guide for local leaders, 2017. 1

[21] Eduardo Blancas Reyes, Jennifer Helsby, Katharina Rasch, Paul van der Boor, Rayid Ghani, Lauren Haynes, and Edward P Cunningham. Early detection of properties at risk of blight using spatiotemporal data. 1

[22] SaoYan. Learntopayattention. https://github.com/SaoYan/LearnToPayAttention, 2018. 5, 6

[23] Jo Schlemper, Ozan Oktay, Liang Chen, Jacqueline Matthew, Caroline Knight, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention-gated networks for improving ultrasound scan plane detection. *arXiv preprint arXiv:1804.05338*, 2018. 2

[24] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2

[25] Paul Hongsuck Seo, Zhe Lin, Scott Cohen, Xiaohui Shen, and Bohyung Han. Hierarchical attention networks. *arXiv preprint arXiv:1606.02393*, 2, 2016. 2

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3

[27] Antje Thiele, Erich Cadario, Karsten Schulz, Ulrich Thonnessen, and Uwe Soergel. Building recognition from multi-aspect high-resolution insar data in urban areas. *IEEE Transactions on Geoscience and Remote Sensing*, 45(11):3583–3593, 2007. 2

[28] Michelle M. Thompson. The city of new orleans blight fight: using gis technology to integrate local knowledge. *Housing Policy Debate*, 22(1):101–115, 2012. 1

[29] Tzutalin. labelimg. https://github.com/tzutalin/labelImg, 2015. 3

[30] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017. 2

[31] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 2

[32] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 420–435, 2018. 2

[33] Matthias Zeppelzauer, Miroslav Despotovic, Muntaha Sakeena, David Koch, and Mario Döller. Automatic prediction of building age from photographs. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 126–134, 2018. 2

[34] Wei Zhang and Jana Košecká. Hierarchical building recognition. *Image and vision Computing*, 25(5):704–716, 2007. 2

[35] Bo Zhao, Xiao Wu, Jiashi Feng, Qiang Peng, and Shuicheng Yan. Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia*, 19(6):1245–1256, 2017. 2, 3, 4