

# Bounding Boxes Are All We Need: Street View Image Classification via Context Encoding of Detected Buildings

Kun Zhao, Yongkun Liu, Siyuan Hao, Shaoxing Lu, Hongbin Liu, Lijian Zhou

**Abstract**—Street view images classification aiming at urban land use analysis is difficult because the class labels (e.g., commercial area), are concepts with higher abstract level compared to the ones of general visual tasks (e.g., persons and cars). Therefore, classification models using only visual features often fail to achieve satisfactory performance. In this paper, a novel approach based on a “Detector-Encoder-Classifier” framework is proposed. Instead of using visual features of the whole image directly as common image-level models based on convolutional neural networks (CNNs) do, the proposed framework firstly obtains the bounding boxes of buildings in street view images from a detector. Their contextual information such as the co-occurrence patterns of building classes and their layout are then encoded into metadata by the proposed algorithm “CODING” (Context encODing of Detected buildINgS). Finally, these bounding box metadata are classified by a recurrent neural network (RNN). In addition, we made a dual-labeled dataset named “BEAUTY” (Building dEtection And Urban funcTional-zone portraYing) of 19,070 street view images and 38,857 buildings based on the existing BIC\_GSV [1]. The dataset can be used not only for street view image classification, but also for multi-class building detection. Experiments on “BEAUTY” show that the proposed approach achieves a 12.65% performance improvement on macro-precision and 12% on macro-recall over image-level CNN based models. Our code and dataset are available at <https://github.com/kyle-one/Context-Encoding-of-Detected-Buildings/>

**Index Terms**—Street view images classification, context encoding, building detection, urban land use classification, urban functional zone, RNN.

## I. INTRODUCTION

URBAN land use records how people use the land with social-economic purposes, such as residential, commercial, and recreational purposes [2]. Land use classification using satellite images have been extensively studied in remote sensing community. With the rise of geo-data commercial services (e.g., Google maps) and crowdsourced projects (e.g., OpenStreetMap) [3], urban spatial data of different modalities are used [4]. As their representative, Google street view (GSV) [5] provides abundant street-level details which have been increasingly used in urban land use classification. Street view images are accurately geo-located, updated regularly,

*Corresponding author: Lijian Zhou.*

Kun Zhao, Yongkun Liu, Siyuan Hao, Shaoxing Lu and Lijian Zhou were with the School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266520, China. E-mail: sterling1982@163.com, YongkunLiu.mail@gmail.com, lemonbananan@163.com, 2445252341@qq.com, zhoulijian@qut.edu.cn

Hongbin Liu was with the BIM Research Center, Qingdao Research Institute of Urban and Rural Construction, Qingdao 266033, China. E-mail: binbin\_sky@163.com

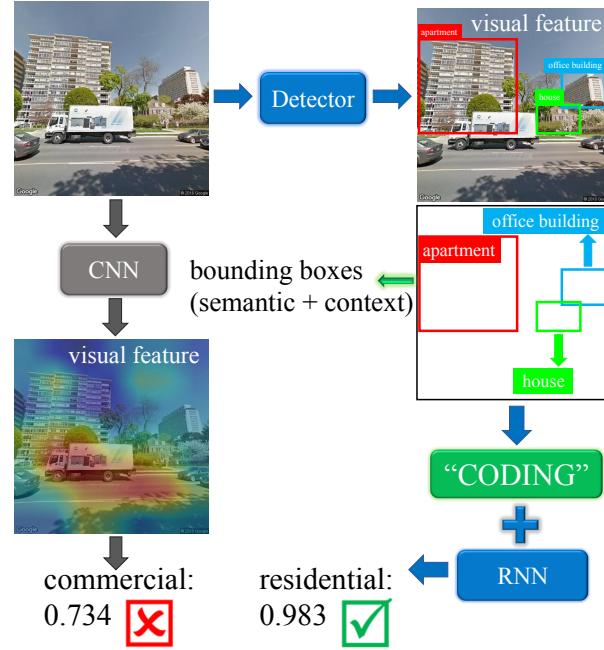


Fig. 1. The main idea of this paper. Left: A common image-level CNN based model. Because of the high-level abstractness of the class labels, the visual features learned by CNN are inaccurate (see the heatmap of the last feature layer), which led to wrong predictions. Right: By using a trained detector, the proposed approach obtained bounding boxes of buildings in the input image, which contain semantic labels and their context information such as the co-occurrence patterns and layout. Correct results were obtained by encoding these information using the proposed “CODING” algorithm and an RNN.

easy and free to access. Moreover, they contain richer visual information which makes it easier to be distinguished (see Fig. 2). Therefore, visual models that perform well in common computer vision tasks, such like CNNs have recently been widely used to extract visual features of street view images for urban land use and urban functional zone analysis [1], [2], [6]–[12]. However, the performance so far has been less than satisfactory partly due to the high-level abstractness of urban land use labels, which makes it hard to represent the concepts directly using visual features. In addition, street view images contain many of the same visual elements (e.g., sky and ground) which interfere with distinguishing different usages of land. When using the whole images directly, the most distinguishable visual elements are underutilized.



(a) A religious area locates round 51.022962, -114.08326.



(b) A residential area locates round 51.029009, -114.07783.

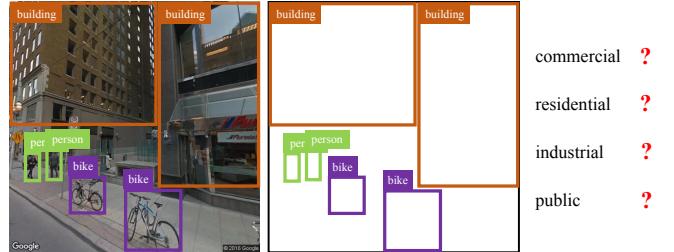
Fig. 2. Areas of different land use with similar looking from overhead view but distinguishable looking from street view.

### A. Motivation

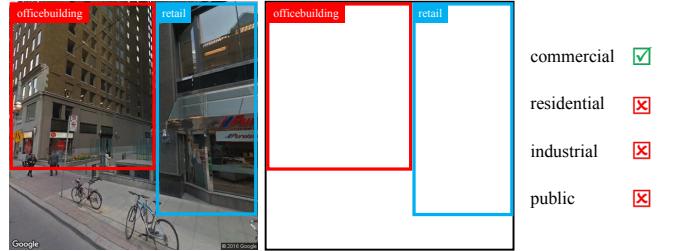
We consider street view image classification as a fine-grained outdoor scene analysis problem. The keys of this task are, firstly, acquiring the most significant objects in street view images aiming at land use, and then, effective modeling of their contextual information. Based on the above viewpoints, a “Detector-Encoder-Classifier” framework is proposed to replace the common CNN based architecture, as shown in Fig 1.

On the first point, significant objects change with specific tasks. Buildings are the main places where people engage in social and economic activities. Urban functional zones also consist mostly of buildings of different categories. Therefore, individual buildings with fine-grained labels should be considered as “significant objects” in street view images for task of urban land use and functional zone analysis. Fig. 3 demonstrate the importance of “significant objects”. Unfortunately, existing open datasets with outdoor scene for common visual tasks [13]–[16] and specific visual tasks (e.g., autonomous driving [17], [18]) are all lack of systematic, fine-grained class definition for buildings. As a milestone work for street view image classification, BIC\_GSV [1] classifies individual buildings into 8 categories. However, its image-level annotation may cause ambiguity when a street view image contains multi-class buildings. In fact, currently there is no dataset using object-level annotations of fine-grained multi-class buildings for street view images.

On the second point, the dominant model for street view image classification is CNN. Recent works either use CNNs directly for end-to-end image-level classification from the same source [1], [6], [8], or CNNs with a two-stream network structure to fuse visual features of images from different sources [2], [7], [9], [10]. As we mentioned, classifying high-level abstract labels directly using visual features may lead to performance bottlenecks. To break the bottlenecks, not only “visual semantics” is needed, but also “visual syntax”. The



(a) A common object-level label system without subclass of “building”.



(b) A land-use oriented label system which takes subclasses of “building” as annotated objects.

Fig. 3. Comparison between common label system and land-use oriented label system in object-level. The subclasses of building enable us to easily distinguish urban functional zones.

former can be obtained by encoding the visual features. The latter should be learned from the context relations of formers.

### B. Contributions

The contributions of the paper lie in three aspects as follows.

- Based on BIC\_GSV, a dual-labeled dataset named **“BEAUTY”** (**B**uilding **E**detection **A**nd **U**rban **F**unctional-zone **po**rtra**Y**ing) with a 19,070 street view images and 38,857 individual buildings by combining automatic labels acquisition from OpenStreetMap (OSM) and expert annotation. It can be used not only for street view image classification aiming at urban land use analysis, but also for multi-class building detection. We also provide baselines for image classification and object detection running on this dataset.
- Based on BEAUTY, a **“Detector-Encoder-Classifier”** framework is proposed to replace the common CNN based architecture. As shown in the right column of Fig. 3(b), without “looking” at the whole image, our approach can infer the land use by only using the bounding boxes of detected buildings. In our approach, object detector is regard as a plug-and-play module that can be arbitrarily replaced, which allows the performance of our approach to easily improve synchronously with the improvement of object detection technology.
- We explored the effect of co-occurrence pattern of multi-class buildings and, further, their spatial layout on urban functional zone analysis. Based on this, we proposed **“CODING”** (**C**ontext enc**O**ding of **D**etected build**ING**s) algorithm to encode the contextual information of bounding boxes into metadata which make it easier to further encoding and classifying using RNN or other models.

### C. Section Arrangement

The rest of the paper is organized as follows. In Section II, we review related work on land use classification using street-level images and current research progress on scene context modeling. Section III introduces our dataset “BEAUTY”. The proposed approach is expatiated in Section IV. Section V shows the experimental setup, results and discussions. Section VI concludes the paper.

## II. RELATED WORK

Urban land use classification has been a growing research field as more data from different sources are available. For example, satellite and aerial images data have been mostly used by the remote sensing community [19], while street-level images were mainly studied by the computer vision community [4]. In the latter, social media images and street view images are the two main sources. Both of them are often referred to a scene analysis problem. In this section, we briefly review the research progress on land use classification using street-level images and context modeling for scene analysis.

### A. Land Use Classification Using Social Media Images

Leung and Newsam [20] first used social media images from Flickr for land use classification. They used the bag of visual words (BOVW) with a soft-weighing scheme to represent image features and then classified them into 3 categories with support vector machine (SVM). Zhu and Newsam improved Leung’s work by using two groups of Flickr images: indoor and outdoor [21], and replacing BOVW features with pre-trained CNN features [15]. Antoniou et al. [22] extracted geo-tagged images from Flickr, Panoramio and Geograph for an area of London, and discussed their usefulness for land use classification. Based on Antoniou’s work, Tracewski et al. [23] used Places205-AlexNet [15] to classify social media images with volunteered geographic information (VGI) for land use classification. Zhu et al. coupled images from Google Places and Flickr with a two-stream CNN to predict the land use [7]. By using ResNet101 as backbone of each branch, they reported 49.54% classification accuracy on 45 categories. Hoffmann et al. [24] classified building instance into 5 land use categories by training a VGG16 using Flickr images.

Social media images provide more street-level details for land use classification. However, they also have shortcomings. First of all, they are often not accurately georeferenced. What’s more, they usually portray highly personalized content (e.g., touristic viewpoints, selfies or zoomed objects) from a subjective, fickle perspective, rather than urban objects from a relatively objective, fixed perspective. Last but not least, they tend to cover the city unevenly (e.g., most images are taken in touristic areas). These problems make such street-level images less suitable for reliable urban land use mapping.

### B. Land Use Classification Using Street View Images

Services like Google Street View (GSV) make it is possible to acquire street-level images with urban objects shot from

a relatively objective perspective, which are accurately geolocated, updated regularly and densely available in many cities all over the world. Recently GSV is being increasingly used in land use classification. Movshovitz et al. [26] used CNN to classify store fronts into 13 business categories from single GSV images. Kang et al. [1] classified urban buildings into 8 categories using GSV images with labels from OSM. Their model predicts one label for each image corresponding to one urban building. Srivastava et al. fused multiple GSV images of a building using a Siamese-like CNN [25] and showed an overall accuracy of 62.52% on 16 OSM label prediction. Noticing that each land use category is made of different objects present in a set of images, they then extended their approach to multi-label prediction [6] to avoid the ambiguity caused by single-label image classification.

Researchers also try to fuse street view images with overhead images by multi-modal strategies. Combination of both modalities was initially used in image geo-localization. Lin et al. [27] matched HRO from Bing Maps with street view images from Panoramio by using four handcrafted features and adding land cover features as the third modality. To extend their approach, they used a Siamese-like CNN to learn deep features between GSV images and 45-degree oblique aerial images [28]. Workman et al. [29] fused overhead images and GSV images by an end-to-end deep network which outputs a pixel-level labeling of overhead images for three different classification problems: land use, building function and building age. They reported a top-1 accuracy of 77.40% and 70.55% for land use classification task on Brooklyn and Queens datasets respectively. Zhang et al. [30] combined airborne light detection and ranging (LiDAR) data, HRO and GSV images for land use classification. In their study, thirteen parcel features were chosen as input variables in a Random Forest classifier which achieves an average accuracy of 77.50%. Cao et al. [2] used images from Bing Maps and GSV for land use segmentation with a two-stream encoder and one decoder architecture which evolved from SegNet [31]. Hoffmann et al. [9] used a two-stream CNN model for building functions classification. They predicted four class labels namely commercial, residential, public and industrial for overhead images by fusing deep features of overhead images and street view images. Their model increases the precision scores from 68% to 76% with a decision-level fusion strategy. Srivastava et al. [10] extend their early work [25] to a multi-modal strategy by leveraging the complementarity of overhead and street-level views. They deal with the situation of missing overhead imagery by using canonical correlation analysis (CCA) based on their two-stream CNN model. By using VGG16 as the backbone, their model achieves an overall accuracy of 73.44% and an average accuracy of 70.30%.

Although the usage of multi-modal strategies gets better results to some extent, the performance so far has been less than satisfactory partly due to the high-level abstractness of urban land use labels which were hard to be abstracted directly using visual features. To break the bottlenecks, some new point of view is needed.

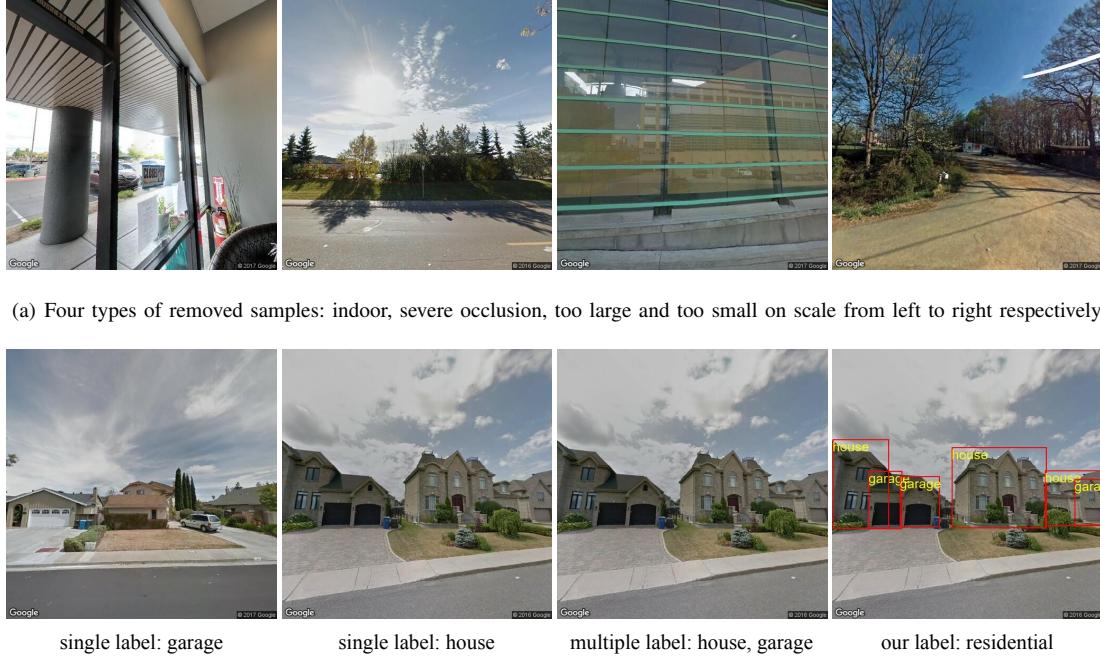


Fig. 4. The improvements made by BEAUTY over BIC\_GSV.

### C. Context Modeling for Scene Analysis

Image context contains a wealth of information about how objects and scenes are related. Cognitive science studies [32], [33] have shown the importance of contextual information in human visual recognition. Typical contextual information including global context [34], visual context [35], object co-occurrence [36] and layout [37], are now exploited to improve the performance of various visual tasks. Pathak *et al.* [38] proposed a context encoder to generate the contents of an arbitrary image region conditioned on its surroundings. Choi *et al.* [39] present a graphical model that combines different sources of context information to detect out-of-context objects and scenes. Izadinia *et al.* [40] encoded the scene category, the context-specific appearances of objects and their layouts to learn scene structures. Chien *et al.* [41] built a CNN to predict the probability of observing a pedestrian at some location in image. Wang *et al.* [42] used a variational auto-encoder to extract the scale and deformation of the human pose and thus predict opportunities of interaction in a scene. Qiao *et al.* [43] proposed an encoder-generator model that encodes the properties of input objects and generates a scene layout representing the scene context. We consider street view image classification as a fine-grained outdoor scene analysis problem. The proposed context encoder will be detailed in Section IV.

### III. DATASET

As we mentioned in Section I-A, currently there is no dataset using object-level annotations of fine-grained multi-class buildings for street view images. Most existing street view datasets use the single-label image-level annotation which contains only global semantics but no descriptions

of content or context. Srivastava *et al.* [25] used a multi-label image-level annotation dataset namely “BAG”<sup>1</sup> which contains object co-occurrence information that could be used to describe contextual relations of the image scene to some extent. However, the labels are for individual buildings such as “office” and “shop”, which lack global semantics of the land use. Furthermore, image-level annotation cannot provide spatial information of objects (e.g., size and position). Thus it contains no richer context information such as layout.

To explore the context relations between street view scene and urban objects in it, a street view image dataset with a dual-label system is made based on the existing BIC\_GSV dataset [1]. On one hand, each image has a land use label to describe the functional zone it portrays, such as “commercial”. On the other hand, each urban object (mostly individual building) in the image is annotated by a bounding box with an object-level label such as “retail”. Thus, the proposed dataset named “BEAUTY” can be used both in land use classification task and in individual building detection task.

BIC\_GSV obtained geo-tagged GSV images located over several cities of the US and Canada (e.g., Montreal, New York City and Denver) and their associated ground truth building labels extracted from OSM. BEAUTY makes the following improvements over BIC\_GSV.

- The remaining invalid samples are further removed. Although BIC\_GSV has removed some outliers with VGG16 trained on Places2 [15], some invalid samples were still found during the manual inspection. As shown in Fig. 4(a), we further remove four types of remaining

<sup>1</sup><https://business.gov.nl/regulation/addresses-and-buildings-databases/>

TABLE I  
THE CORRESPONDENCE BETWEEN THE PROPOSED FOUR LAND USE CATEGORIES AND OSM, LBCS LABELS.

OpenStreetMap Land Use Tag	Proposed Urban Land Use Classes	LBCS Function Dimension
residential	residential	1000: residence or accommodation functions
garages	residential	-
commercial	commercial	2000: general sales or services
retail	commercial	2000: general sales or services
cemetery	public	-
recreation ground	public	6000: education, public admin., health care, religious and other institutions
religious	public	6000: education, public admin., health care, religious and other institutions
village green	public	6000: education, public admin., health care, religious and other institutions
-	public	4000: transportation, communication, information, and utilities
industrial	industrial	3000: manufacturing and wholesale trade

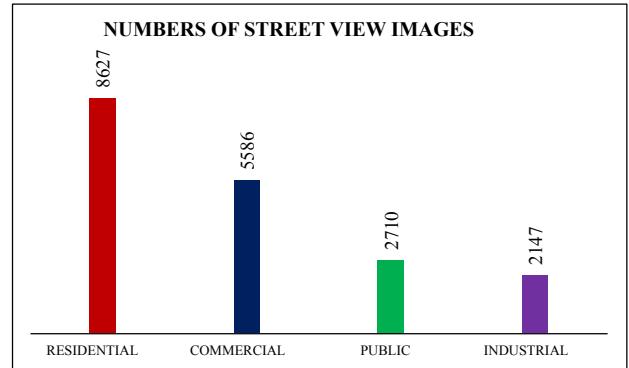
invalid samples: indoor, severe occlusion, too large and too small on scale.

- Object-level annotations are given for each building in an image. In combination with the building labels<sup>2</sup> automatically obtained from OSM, we manually annotate each individual building in each image under the guidance of architecture experts. In object-level annotations, we use the 8 class labels used in BIC\_GSV, namely *apartment*, *church*, *garage*, *house*, *industrial*, *office building*, *retail* and *roof*. Object-level annotations avoid the ambiguity when buildings in different classes are in the same image and also afford the layout information of buildings in the same scene of land use. An example is shown in Fig. 4(b).
- Image-level labels are further abstracted into land use categories. In combination with the land use labels<sup>3</sup> automatically obtained from OSM and the Land Based Classification Standards (LBCS) Function Dimension with Descriptions<sup>4</sup>, we manually annotate each image under the guidance of urbanist. We fuse the OSM land use labels and LBCS urban function descriptions into 4 highly compact classes namely *commercial*, *residential*, *public* and *industrial*, which have been used in [9], because such a classification has a very high value to urban geography being correlated with socio-demographic parameters such as population density and income. The correspondence between the proposed four land use categories and OSM, LBCS labels is shown in TABLE I.

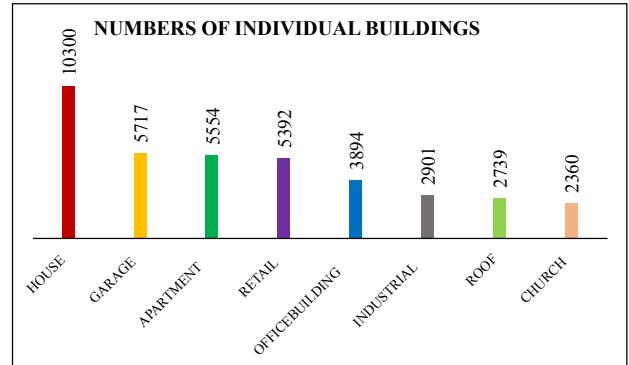
The BEAUTY dataset consists of 19,070 street view images with 38,857 individual buildings. As can be seen from Fig. 5, both the sample distributions of land use classes and building classes are long-tailed, which are in line with the situation in the real world. Fig. 6 shows samples of proposed dataset.

#### IV. PROPOSED APPROACH

Fig. 7 shows the pipeline of proposed approach. The inputs are street view images and the outputs are their predicted land use categories which would be mapped to the geographic information systems according to their geo-location.



(a) Sample numbers of each land use class.



(b) Sample numbers of each building class.

Fig. 5. Sample distributions of BEAUTY.

As we mentioned in Section I-A, the first key of our task is acquiring the most significant urban objects in street view images. The building detector plays the key role to do it. Two off-the-shelf detectors were used in this paper namely Faster R-CNN [44] and Cascaded R-CNN [45]. The detectors were trained using the object-level ground truth of training samples in BEAUTY. The outputs of the detector are bounding boxes of each building with their classes and confidence scores, which would be transferred into metadata through the context encoder. In addition to being an intermediate module of the street view classification task, we also consider building

<sup>2</sup>[https://wiki.openstreetmap.org/wiki/Map\\_Features#Building](https://wiki.openstreetmap.org/wiki/Map_Features#Building)

<sup>3</sup>[https://wiki.openstreetmap.org/wiki/Map\\_Features#Landuse](https://wiki.openstreetmap.org/wiki/Map_Features#Landuse)

<sup>4</sup><https://www.planning.org/lbcs/standards/function/>

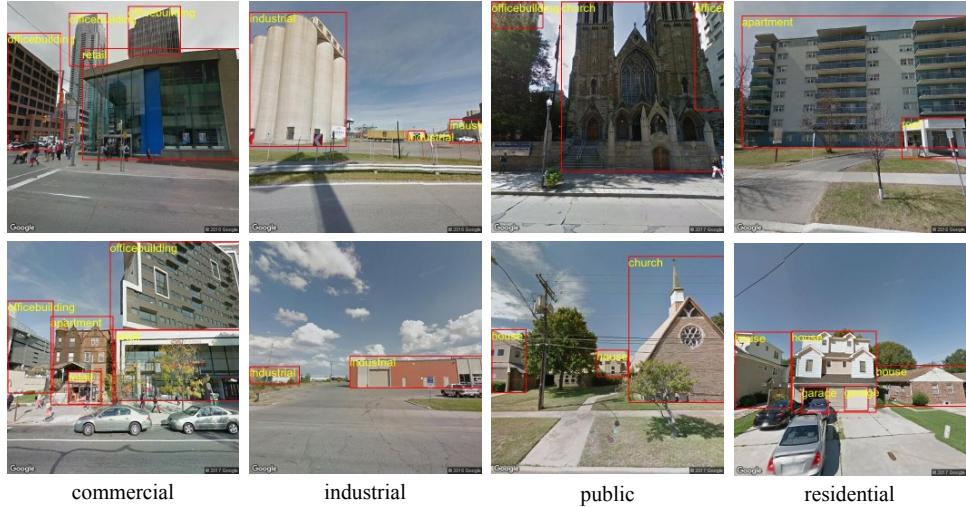


Fig. 6. Samples of BEAUTY: street view images in four land use scenes with different type of buildings.

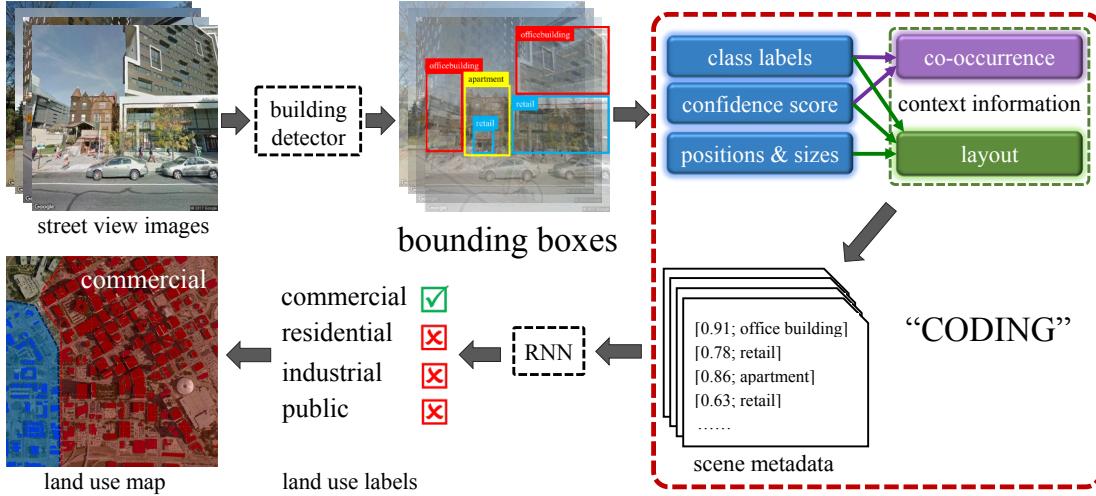


Fig. 7. Pipeline of proposed approach using a “Detector-Encoder-Classifier” framework. The core algorithm “CODING” encodes the input bounding boxes into scene metadata containing contextual information.

detection as a separate task and conduct the corresponding baseline tests. This part will be detailed in Section V-B.

As shown in Fig. 7, two different kinds of contextual information are optional in the proposed “CODING” module, which are co-occurrence patterns and layout.

#### A. Context Encoding Using Object Co-occurrence Only

Outputs of a detector are bounding boxes of detected object regions, each of which consists of the following data: predicted confidence score, class label and position in the image. The class and confidence score are concrete representations of semantic information. We integrate them together into a feature vector with the form of the hot-one vector of predicted class whose none-zero value was replaced by the confidence score. These “semantic vectors” are used directly to the classification task. The position data are often with the form of  $[x_i/W, y_i/H, w_i/W, h_i/H]$  where  $x_i, y_i$  are the coordinates of the top left corner of the detection bounding box,  $w_i, h_i$  are

its width and height, and  $W, H$  are the width and height of the image. The use of position information will be explained in detail in the next part.

The semantic vectors of detections are grouped by image and mapped to a set which is padded to length  $l$  by all-zero vectors. We set  $l = M + m$ , where  $M$  is the max detections of one image in training set and  $m$  is a slack. A set of vectors is obtained without using position data. The set contains only co-occurrence information of different buildings in a scene and could be further encoded and classified.

#### B. Context Encoding Using Building Layout

The absolute position of bounding boxes were not encoded directly because the angle and scale of the building shot in the street view image change dramatically, which makes the features lack of angle and scale invariance. The position vectors are used to compute intermediate variables such as the relative size of bounding boxes and the distance between

them. These intermediate variables help to decide the order of semantic vectors. Thus, the sequence implies the relative layout of buildings and preserves the invariance of angle and scale of features simultaneously. The specific steps of sequence generation are shown in Algorithm 1, where  $\mathbb{B}$  denotes the set of detected bounding boxes  $B_i$  with its hot-one vector of class  $\vec{C}_i$ , predicted confidence score  $p_i$  and its position vector  $[x_i, y_i, w_i, h_i]$ , and  $\mathbb{S}$  denotes the generated sequence.

**Algorithm 1** Sequence generation for layout encoding

```

1: Input: Set of bounding boxes  $\mathbb{B}$ 
2: Output: Sequence of semantic vectors  $\mathbb{S}$ 
3:  $\mathbb{S} \leftarrow \emptyset$ 
4: for  $B_i \in \mathbb{B}$  do
5:    $a_i \leftarrow w_i \times h_i \times p_i$ 
6:    $\hat{x}_i \leftarrow x_i + w_i/2$ 
7:    $\hat{y}_i \leftarrow y_i + h_i/2$ 
8:    $\vec{C}_i^* \leftarrow p_i \times \vec{C}_i$ 
9: end for
10: PUSH  $\vec{C}_0^*$  :  $a_0 = \max\{a_i\}$  INTO  $\mathbb{S}$ 
11: DELETE  $B_0$  :  $a_0 = \max\{a_i\}$  FROM  $\mathbb{B}$ 
12: DELETE  $\vec{C}_0^*$  :  $a_0 = \max\{a_i\}$  FROM  $\{\vec{C}_i^*\}$ 
13: for  $B_i \in \mathbb{B}$  do
14:    $d_i \leftarrow \sqrt{(\hat{x}_i - \hat{x}_0)^2 + (\hat{y}_i - \hat{y}_0)^2}$ 
15: end for
16: ASCENDING_SORT  $\vec{C}_i^*$  BY  $d_i$ 
17: for  $\vec{C}_i^* \in \{\vec{C}_i^*\}$  do
18:   PUSH  $\vec{C}_i^*$  INTO  $\mathbb{S}$ 
19: end for
20: RETURN  $\mathbb{S}$ 
```

To put it simply, we first select the bounding box with highest confidence score and largest size (Line 5) to be the leading box (Line 10), and then ascending sort the rest ones (Line 16) by the centroids distance (Line 14) between them and the leading box. Finally, sequences of vectors with hot-one like form (semantic structure), and their order (syntax structure) constitute the scene metadata.

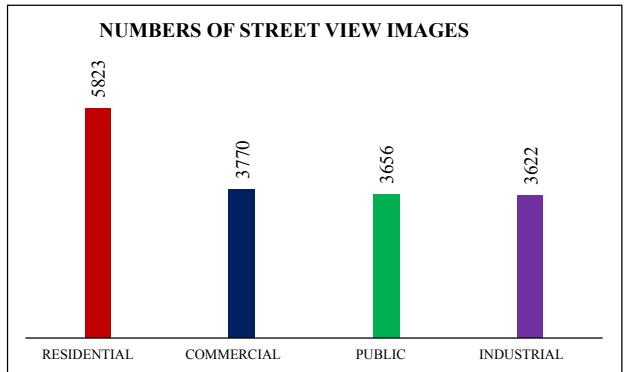
To further encoding and classifying the metadata obtained from “CODING”, two RNN architectures are used, namely last-layer-concatenated single-directional RNN and all-concatenated bidirectional RNN (BRNN) [46], both with two hidden layers. The inputs of RNNs are the semantic vectors with the size of 8, representing the detected bounding boxes of 8 class buildings in an image. The size of a hidden layer neuron  $h_i$  is 16 and the dimension of the parameter matrix  $\mathcal{A}$  in the first hidden layer is  $16 \times (16 + 8)$ . The basic units follow a simple RNN structure. It can also be replaced by a gated recurrent unit (GRU) [47] or a long short-term memory (LSTM) [48] unit, which will be compared in Section V-C.

In the first architecture, all hidden neurons in the last layer are concatenated. In order to reduce the weights of zero vectors generated by padding, vectors of the input metadata should be arranged in reverse order. The concatenation layer ensures that no feature of a single bounding box would be forgotten by the directionality of the RNN. Well in the BRNN architecture, all neurons in both hidden layers are finally concatenated to connect to a full connection (FC) layer and output the

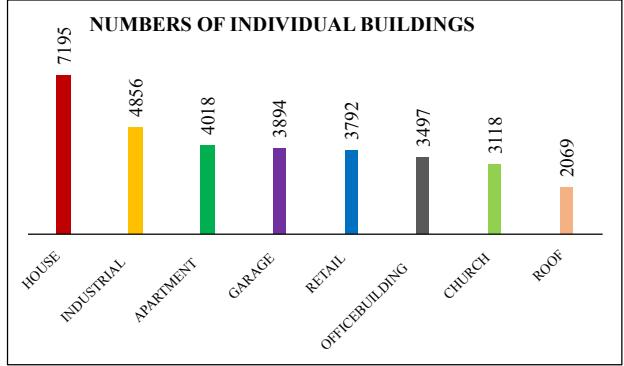
predicted probability of four classes of land use scenes after a softmax operation.

## V. EXPERIMENTS AND ANALYSIS

In order to verify the validity of the proposed approach, we ran it on BEAUTY dataset which has been introduced in detail in Section III. In this section, we will first conduct baseline tests for the tasks of street view image classification and building detection on this dataset, then compare the performance of proposed approach with baseline, and draw some useful conclusions through analysis.



(a) Sample numbers of each land use class.



(b) Sample numbers of each building class.

Fig. 8. Training sample distributions of land use classes and building classes after rebalance.

### A. Data Preparation and Experimental Setup

We randomly selected 75% of the samples from each category as the training/validation set and the remaining 25% as the test set. The training/verification set was then randomly divided into training set and verification set according to the ratio of 9:1. From Fig. 5 we can learn that there is a class imbalance problem [49] in BEAUTY. To reduce the impact of class imbalance and achieve better performance, we carried out class rebalance for the training samples using a random minority oversampling strategy. Specifically, the samples of *public* and *industrial* were expanded by 2 times and 2.5 times respectively. Since all training samples will be flipped randomly in the horizontal direction in the data augmentation stage before training, only random copy is needed for minority

oversampling, which is a common strategy in the industry. We also tried random geometry transformation and random color jittering [50] for training data augmentation. Unfortunately, the performance is not as good as random horizontal flip considering both the tasks of street view image classification and building detection. The sample distributions after rebalance are shown in Fig. 8. The total number of training images and individual buildings are 16,871 and 32,439 respectively after class rebalance before data augmentation. Compare Fig. 8 with Fig. 5 we can learn that not only the image-level samples of land use classes are rebalanced, but also the object-level samples of building classes.

All experiments are based on the same hardware and software conditions as follows. GPU: GeForce GTX 1080  $\times$  2; OS: Ubuntu 18.04.3 LTS; CUDA Version: 10.0.130; PyTorch Version: 1.4.0 for cu100; TorchVision Version: 0.5.0 for cu100. We set the number of RNN input  $l = 25$ , which ensures that most bounding boxes are involved and avoids too much zero vectors. The pre-trained models and the training hyperparameters will be presented in detail in later sections. All the results were averaged after 10 runs.

### B. Baselines

To facilitate the evaluation of model performance on street view image classification and building detection on BEAUTY, we selected the corresponding baseline models for both tasks. Considering the leading role of CNN-based end-to-end model in image classification and object detection tasks in recent years, we chose the most representative and most widely used ResNet [51] model and two detectors based on it as the baseline models.

1) *Baseline Test for Street View Image Classification*: For the task of street view image classification, ResNet50 and ResNet101 are selected as the candidate baseline models. We finetuned the pre-trained models<sup>5</sup> on BEAUTY with the learning rate of 0.01, which was multiplied by a factor of 0.1 after every 10 epochs. The training was pursued for 100 epochs with Adam [52] as an optimizer. The training and validation losses are drawn in Fig. 9.

Although the training set has been rebalanced, serious class imbalance remained in the test set of BEAUTY. Therefore, as a commonly used metric for classification, overall accuracy is not suitable for the evaluation on our dataset. Instead, the macro-average of the per-class metrics are used, namely macro-precision (M-P), macro-recall (M-R) and macro F1-Score (M-F1) [53]. As shown in TABLE II, the performance of ResNet50 beats ResNet101 in all macro-average metrics. Thus, ResNet50 is selected as the baseline model for street view image classification on BEAUTY. The confusion matrices in Fig. 10 also show that ResNet50 performs better than ResNet101 in all other categories with the exception of *public*.

2) *Baseline Test for Building Detection*: For the task of building detection, Faster R-CNN [44] and Cascaded R-CNN [45] with the backbone of ResNet50 and ResNet101 are selected as the candidate baseline models. We finetuned the

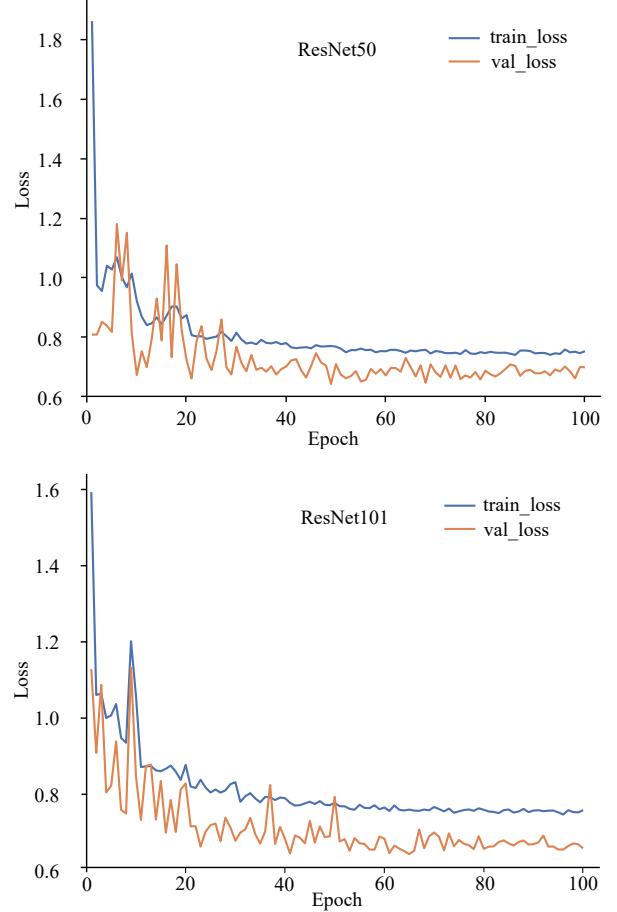


Fig. 9. The training and validation losses of ResNet50 and ResNet101.

TABLE II  
PERFORMANCES OF RESNET50 AND RESNET101 IN PERCENTAGE TERMS.

Models	M-P	M-R	M-F1
ResNet50	<b>69.16</b>	<b>68.94</b>	<b>69.05</b>
ResNet101	67.48	68.87	68.17

pre-trained models<sup>6</sup> of MMDetection [54] using their default hyperparameters on BEAUTY. Part metrics used in COCO 2020 Object Detection Task<sup>7</sup> are used and extended as our metrics for building detection, which are “average precisions” (AP) at “intersection over union” (IoU) of different values. For example, AP<sub>IoU=.50:.05:.95</sub> refers to the AP of all classes when the detections and ground truth bounding boxes were matching according to the least IoU value to be 0.5 to 0.95. It is a relatively strict metric, because a high IoU lower limit represents a high requirement for the position accuracy of the prediction box. In contrast, AP<sub>IoU=.50</sub> is a metric with relatively loose requirement for position accuracy. In order to compare the effectiveness of visual feature extraction between the detector and the end-to-end classifier, we extend this set of metrics to AP<sub>IoU=.00</sub>, which means that the location of

<sup>5</sup><https://pytorch.org/docs/stable/torchvision/models.html>

<sup>6</sup>[https://mmdetection.readthedocs.io/en/latest/model\\_zoo.html](https://mmdetection.readthedocs.io/en/latest/model_zoo.html)  
<sup>7</sup><https://cocodataset.org/#detection-eval>

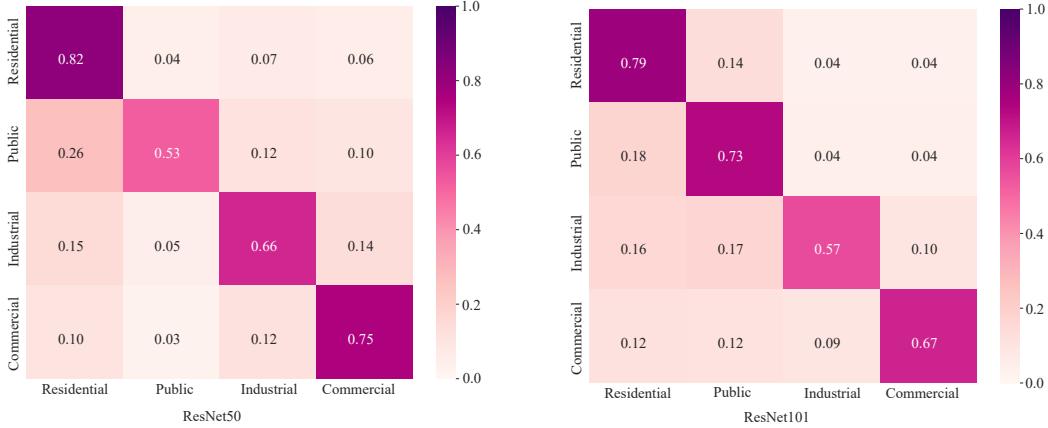


Fig. 10. The confusion matrices of ResNet50 and ResNet101.

an object detection is not considered, but only whether its category is correct for evaluation. In essence, this metric is equivalent to the macro-precision of a multi-label classifier.

The performances of candidate detectors with candidate backbones on  $\text{AP}^{\text{IoU}=.50:.05:.95}$  to  $\text{AP}^{\text{IoU}=.00}$  are shown in TABLE III, where Fa-50 refers to Faster R-CNN with the backbone of ResNet50, and Ca-101 refers to Cascaded R-CNN with the backbone of ResNet101. As the metrics became looser, the detectors scored higher. When consider the accuracy of bounding box position, Ca-101 was the best detector. In contrast, Fa-50 becomes optimal when only the accuracy of the class prediction is considered. We select Ca-101 to be the default detector for our system because our approach encodes the layout of buildings by using both the class and position information. If not specified, the detectors used in the subsequent experiments are all Ca-101. TABLE III is also considered as the baseline for an independent visual task namely multi-class building detection on BEAUTY.

TABLE III  
PERFORMANCES OF CANDIDATE DETECTORS IN PERCENTAGE TERMS.

Detectors	$\text{AP}^{\text{IoU}=.50:.05:.95}$	$\text{AP}^{\text{IoU}=.75}$	$\text{AP}^{\text{IoU}=.50}$	$\text{AP}^{\text{IoU}=.00}$
Fa-50	46.09	50.71	69.70	<b>79.33</b>
Fa-101	46.11	50.73	69.42	79.01
Ca-50	48.72	53.24	<b>70.21</b>	79.11
Ca-101	<b>48.92</b>	<b>53.88</b>	70.13	79.10

Compared with the best M-P (69.16) in TABLE II, the best AP (79.33) in TABLE III is significantly improved with the same backbone CNN architecture (ResNet50). Although this comparison is not rigorous, we can still roughly observe that the individual buildings in images are easier to be visually characterized and abstracted more effectively than the whole street view image by the same visual feature extractor. This conclusion is the cornerstone of the superiority of our approach over image-level end-to-end CNN models. More details about the effectiveness of visual extraction are presented in Section V-D1.

### C. Comparison of Different Settings

After Ca-101 is selected as the default detector, Section IV provides two contextual encoders, namely the co-occurrence encoder and the layout encoder and two RNN network structures, namely single-directional RNN and bidirectional RNN, with three basic network units, namely simple-RNN unit, GRU and LSTM unit. In the following sections, we will discuss these options and try to find the best combination.

1) *Performance of Co-occurrence Encoder:* TABLE IV shows the performance of the co-occurrence encoder combined with different RNN classifiers. For classifiers using simple-RNN and LSTM units, the single-directional structures are better than the bidirectional ones. The reverse is true when GRU is used. Simple-RNN units in both structures clearly outperform the others, which makes it to be the default network unit. The best combination belongs to simple-RNN units with single-directional structure, which is regarded as the best classifier for co-occurrence encoder.

TABLE IV  
PERFORMANCES OF CO-OCCURRENCE ENCODER IN PERCENTAGE TERMS.

Combinations	M-P	M-R	M-F1
simple-RNN+single-directional	<b>81.47</b>	<b>80.53</b>	<b>81.00</b>
simple-RNN+bidirectional	81.13	80.20	80.66
GRU+single-directional	80.43	79.22	79.82
GRU+bidirectional	80.57	79.24	79.90
LSTM+single-directional	80.85	79.50	80.17
LSTM+bidirectional	80.50	79.39	79.94

2) *Performance of Layout Encoder:* TABLE V shows the comparison between co-occurrence encoder and layout encoder combined with single-directional and bidirectional structure using simple-RNN units. Layout encoder clearly beats co-occurrence encoder, indicating that the spatial arrangement of the building reflects a certain structural context and is useful for distinguishing street view images with different types of land use. For layout encoders, the structure of RNN does not matter much. This also indicates that the spatial arrangement of buildings has a certain robustness for distinguishing different land use scenes.

TABLE V  
PERFORMANCES OF CO-OCCURRENCE AND LAYOUT ENCODER IN PERCENTAGE TERMS.

Combinations	M-P	M-R	M-F1
co-occurrence+single-directional	81.47	80.53	81.00
co-occurrence+bidirectional	81.13	80.20	80.66
layout+single-directional	81.66	<b>81.02</b>	81.34
layout+bidirectional	<b>81.81</b>	80.94	<b>81.37</b>

3) *RNN Training Using Ground Truth Bounding Boxes:* From Fig. 7 we know that our RNN is trained by the bounding boxes output from detectors. Why don't we use the ground truth bounding boxes to train the RNN and use outputs of detectors during test stage? TABLE VI shows the comparison between using and not using the ground truth bounding boxes during training stage. The results are disappointing. The “standard answer” seems to be helpless might due to the mismatch during training and test stage.

TABLE VI  
COMPARISON BETWEEN USING AND NOT USING THE GROUND TRUTH BOUNDING BOXES FOR TRAINING.

Training Combinations	M-P	M-R	M-F1
co-occurrence+ground truth	77.76	71.65	74.58
co-occurrence+Ca101 best	81.47	80.53	81.00
layout+ground truth	80.03	80.93	80.48
layout+Ca101 best	<b>81.81</b>	<b>80.94</b>	<b>81.37</b>

Comparing the co-occurrence coding and layout coding, it can be found that the influence of training-test mismatch on the latter (0.89% in M-F1) is significantly lower than that on the former (6.42% in M-F1). This observation once again demonstrates the robustness of spatial structure. So far, the optimal performance of the proposed approach is generated by layout coding combined with bidirectional RNN structure.

#### D. Comparison with Baseline

Finally, the upper limit and the optimal performance of proposed approach and baseline are compared in TABLE VII. In Section V-C3, the poor performance has been shown when using ground truth bounding boxes as training samples but the outputs of a detector as test samples, due to the training-test mismatch. How about we use the ground truth bounding boxes also in the test stage? It is impossible for a classification system to know some intermediate results of test samples in advance (e.g., ground truth bounding boxes of buildings in the test sample images), but the hypothesis could help us to find out the performance upper limit of the proposed encoder-classifier system. Upper limit means that a perfect detector is used, whose outputs during training and test stages are ground truth bounding boxes which make the proposed encoder-classifier system perform best.

The M-F1 of the perfect detector shows a 12.45% higher than the current optimal combination of the proposed approach, which means that our approach has a lot of room

TABLE VII  
THE UPPER LIMIT, PROPOSED APPROACH AND BASELINE.

Models	M-P	M-R	M-F1
layout+perfect detector	95.54	92.15	93.82
layout+Ca101 best	<b>81.81</b>	<b>80.94</b>	<b>81.37</b>
base line: ResNet50	69.16	68.94	69.05

to improve with the progress of object detection. On the other hand, the M-F1 of the proposed approach presents a 12.32% higher than the baseline (ResNet50 image classification model), which is a significant improvement. More details could be obtained in the confusion matrices of the four classes of land use scenes (Fig. 11). The category with the most room for improvement of the proposed method is industrial (16%). While the category with the most improvement over the baseline is public (39%). These are discussed in more details in the following parts.

1) *Typical Case Analysis in Visual Feature Extraction:* In this part, we try to explain why the significant improvement by proposed approach over common image-level end-to-end visual models in terms of the effectiveness of visual feature extraction. In order to demonstrate the effectiveness of visual features more intuitively, we use visual feature heatmap [55], which is often used for interpretability analysis of neural networks. For baseline (ResNet50), feature maps before the last average pooling layer are used to generate the heatmaps. The regions that contribute to the prediction of each class are marked by warm color regions in the heatmap of the class. For the proposed approach, the approximate heatmaps are generated by the outputs of detector (Fa-50). For each bounding box  $b_i$ , its approximate heatmap is assumed to be a two-dimensional Gaussian distribution that described by (1), where  $T_{x,y}$  is the temperature of point  $(x, y)$  in an image,  $w_i$ ,  $h_i$  are the width and height of  $b_i$  and  $(x_{i0}, y_{i0})$  are the center coordinates of  $b_i$ . Since the detector was not directly used for scene classification, we overlaid and normalized the heatmap of each detection to show the regions that were potentially helpful for the final classification. Heatmaps of typical cases are shown in Fig. 12.

$$T_{x,y} = \frac{1}{\pi\sqrt{w_i h_i}} \exp \left\{ -2 \left[ \frac{(x - x_{i0})^2}{w_i^2} + \frac{(y - y_{i0})^2}{h_i^2} \right] \right\} \quad (1)$$

We chose typical cases from the categories with the most improvement over the baseline namely public (39%) and industrial (5%) to do an in-depth analysis. Fig. 12(a) shows a case of *public* (image ID: *public\_383* in BEAUTY). The activated regions of the heatmap for the correct class (*public*) include large areas of sky and ground. While the activated regions of the heatmap for the prediction (*residential*) do not cover the whole buildings and miss the cross of the church in left, which carries key information about land use. It also contains lots of areas of ground. In contrast, the heat map generated by detector covers all key regions tightly. A similar situation can be clearly demonstrated in the *industrial* example (Fig. 12(b)).



Fig. 11. The confusion matrices of upper limit (left), optimal performance of proposed approach (middle) and baseline (right).

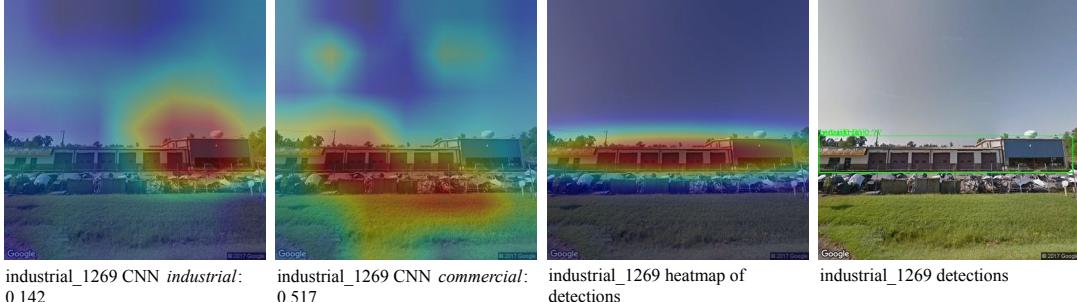
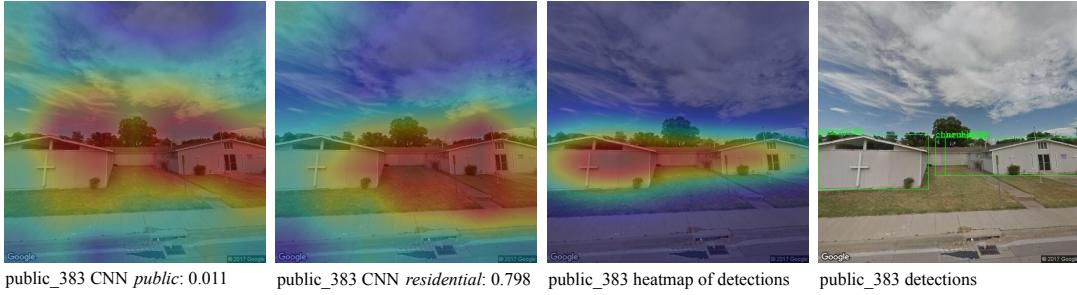


Fig. 12. Heatmaps of the correct class (left 1) and prediction (left 2) by ResNet50 and detections (left 3).

2) *Typical Case Analysis in Context Information Extraction*: Cases in previous part demonstrate that the proposed approach can obtain more effective visual features than CNN-based image-level end-to-end model by using detectors specially trained for buildings. Is the good performance of the proposed approach entirely dependent on the detector? In this part, several cases will show that the proposed encoder and RNN-based classifier can obtain correct scene classification according to the learned context information, even if the detector incorrectly predicts the class of some buildings. Outputs of the detector are firstly encoded by the proposed “CODING”. Then a growing number of bounding boxes with high confidence scores have been tampered with as the ones of wrong categories. During this process, the results of the final scene classification are observed all the time. Cases are shown in Fig. 13.

For illustration purposes, only bounding boxes with a confidence score greater than 0.4 are shown. The actual number of detections of the first case “commercial\_5548” is 12, showing only 7 in Fig. 13(a). Parts of the bounding boxes overlap, such as “retail: 0.95” and “office building: 0.56” on the far right, and “office building: 0.56” and “garage: 0.50” next to them. Several bounding boxes are tampered with to the wrong class one by one. When two bounding boxes were tampered with (*office building: 0.56* → *apartment: 0.56*, *office building: 0.89* → *garage: 0.89*), the proposed approach could still make correct prediction about the scene class. When the third bounding box was tampered with (*office building: 0.96* → *house: 0.96*), the prediction jumped from *commercial: 0.82* to *residential: 0.88* without a gradient. In the second case “residential\_5173”, there are some errors in the original detections. The ground truth of the largest bounding box in the lower left corner

(a) A case of *commercial*. Wrong prediction happened after three bounding boxes are tampered with.(b) A case of *residential*. Wrong prediction happened after four bounding boxes are tampered with.

Fig. 13. The prediction of proposed approach keeps correct when the detector makes a small number of wrong predictions (red bounding boxes).

is *garage*, but the detection is *apartment*: 1.00. The ground truth of the small bounding box on the left is *apartment*, but the detection is *garage*: 0.94. This kind of detection errors could cause minor changes to the layout encoding and will be ignored in RNN-based classifier. And since the co-occurrence relationship between building classes does not change, it did not affect the final classification result (*residential*: 1.00). In Fig. 13(b), class of the detections are tampered with one by one from right to left. The prediction jumped from *residential*: 0.98 to *public*: 0.98 after four bounding boxes were tampered with. Some bounding boxes that were not drawn because their confidence scores were less than 0.4 still contributed to the context relationships such as co-occurrence and layout, which resulted in the prediction of scene class being maintained when three detections were tampered with. The last straw was the manipulation of the bounding box with the largest size and confidence score (*apartment*: 1.00 → *church*: 1.00).

The case analysis above can give a glimpse of why the proposed approach achieved better performance over image-level end-to-end CNN models such as ResNet50. The general conclusions are given in Section VI.

#### E. Use on Open World Street View Image Data

To further verify the performance of the proposed approach on an open world data set, land use maps of Calgary are generated using open world GSV images provided by [1]. Land use maps of Calgary based on 6,124 street view images are shown in Fig. 14. Geo-tagged street view images were classified by the proposed approach. The results were then

drawn on CesiumJS<sup>8</sup> according to the geographical locations of the input images. Four land use classes *residential*, *commercial*, *industrial* and *public* are marked by dots of blue, red, purple and yellow respectively. Regions where dots with the same color clustered in a city-scale map are zoomed in to see if the classification is correct.

Generally speaking, the distribution of *residential* area, *commercial* area and *industrial* area in Calgary is relatively balanced. The *commercial* areas (red dots) are relatively concentrated, while the other two are scattered. In Fig. 14, a *commercial* area, an *industrial* area and a *residential* area that was shown as red, purple and blue dots cluster respectively were zoomed in. As can be seen from the zoomed over-head image on the top right, the buildings of *commercial* area are dominated by tall buildings. In the middle right zoomed image, the *industrial* area is dominated by large flat-roofed buildings with low floors, which is a typical feature of the *industrial* area. In zoomed over-head image on the bottom right, the characteristics of *residential* areas are also obvious for a large number of small well-arranged low-rise buildings. The *public* areas represented by yellow dots rarely form clusters.

## VI. CONCLUSION AND FUTURE WORK

As the CNNs gradually show an overwhelming advantage in common visual tasks, various image-level CNN models are increasingly favored in street view image classification in recent years. In this paper, a dataset “BEAUTY” is presented, which can be used for both street view image classification

<sup>8</sup><https://cesium.com/cesiumjs/>



Fig. 14. The city-scale land use classification map of Calgary.

and building detection. We used ResNet50, which performs steadily and well on common visual classification tasks as a baseline model to represent the current mainstream image-level CNN models. However, the macro-precision of the ResNet50 was only 69.16%. After analysing large number of street view image samples, we find that the approaches based on image-level CNN models have the following fatal problems.

- The undifferentiated use of the whole image leads to the extraction of common visual factors that confuse classification.
- Street view image labels for land use classification are often concepts with a high level of abstraction and cannot be described directly and effectively with visual features.

As can be seen from the example in Fig. 12, although CNNs have the ability to extract regions conducive to classification through autonomous learning, these regions are often not accurate when classification labels cannot be directly and effectively described with visual features. In addition, the only use of visual semantics (e.g., the recognition results of objects) can no longer well represent highly abstract land-use concepts, which must be done with context-describing visual syntax. Based on the above considerations, this paper proposes a “Detector-Encoder-Classifier” architecture. Object detectors extract visual features that are more recognizable

by learning the annotations specifically for buildings. The proposed “CODING” method encodes the context relations such as co-occurrence and layout of these highly recognizable visual objects. At last, RNNs are very suitable for accurately classifying the combination patterns of visual elements with structural relations. The proposed approach performs 81.81% on macro-precision, an improvement of 12.56% over the baseline model.

The first row of TABLE VII shows the performance of the proposed approach using a “perfect detector”, which gives the upper limit of the proposed approach and two ideas for improving the performance under the current architecture.

- To achieve the upper limit, better detectors are needed. With the development of object detection, this plug-and play-module can be upgraded continuously.
- To exceed this limit, more powerful context encoders need to be proposed. Self-attention [56] or transformers model [57] might be used.

In addition to models, data from different sources are also an important way to improve the performance. A more accurate description of the land use may be obtained by matching the layout of the building in street view images to one in overhead images.

## ACKNOWLEDGMENT

The authors would like to thank the authors of reference [1] for publishing the BIC\_GSV dataset including city scale GSV images. We would also like to thank Hongbin Liu and Zhiwei He, the experts in architecture and urban planning from the BIM Research Center, Qingdao Research Institute of Urban and Rural Construction for their professional guidance on manual annotation. Thanks to those who participated in manual annotation for building detection: Yu Ma, Shanshan Lin, Ying Guo and Kaixin Li, and who participated in manual annotation for street view image classification: Ying Zhang, Jiaojie Wang, Shujing Ma and Yue Wang. This work has been supported by the National Natural Science Foundation of China (Grant No. 61701272).

## REFERENCES

- [1] J. Kang, M. Körner, Y. Wang, H. Taubenböck, and X. X. Zhu, "Building instance classification using street view images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 44–59, 2018.
- [2] R. Cao, J. Zhu, W. Tu, Q. Li, J. Cao, B. Liu, Q. Zhang, and G. Qiu, "Integrating aerial and street view images for urban land use classification," *Remote Sensing*, vol. 10, no. 10, p. 1553, 2018.
- [3] J. Vargas, S. Srivastava, D. Tuia, and A. Falcao, "Openstreetmap: Challenges and opportunities in machine learning and remote sensing," *arXiv preprint arXiv:2007.06277*, 2020.
- [4] S. Lefèvre, D. Tuia, J. D. Wegner, T. Produt, and A. S. Nassar, "Toward seamless multiview scene analysis from satellite to street level," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1884–1899, 2017.
- [5] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, vol. 43, no. 6, pp. 32–38, 2010.
- [6] S. Srivastava, J. E. Vargas Muñoz, S. Lobry, and D. Tuia, "Fine-grained landuse characterization using ground-based pictures: a deep learning solution based on globally available data," *International Journal of Geographical Information Science*, vol. 34, no. 6, pp. 1117–1136, 2020.
- [7] Y. Zhu, X. Deng, and S. Newsam, "Fine-grained land use classification at the city scale using ground-level images," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1825–1838, 2019.
- [8] L. Ilic, M. Sawada, and A. Zarzelli, "Deep mapping gentrification in a large canadian city using deep learning and google street view," *Plos One*, vol. 14, no. 3, p. e0212814, 2019.
- [9] E. J. Hoffmann, Y. Wang, M. Werner, J. Kang, and X. X. Zhu, "Model fusion for building type classification from aerial and street view images," *Remote Sensing*, vol. 11, no. 11, p. 1259, 2019.
- [10] S. Srivastava, J. E. Vargas-Muñoz, and D. Tuia, "Understanding urban landuse from the above and ground perspectives: A deep learning, multimodal solution," *Remote Sensing of Environment*, vol. 228, pp. 129–143, 2019.
- [11] S. Lobry, D. Marcos, J. Murray, and D. Tuia, "Rsvqa: Visual question answering for remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.
- [12] Q. Liu, S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat v2: feature augmented convolutional neural nets for satellite image classification," *Remote Sensing Letters*, vol. 11, no. 2, pp. 156–165, 2020.
- [13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 740–755.
- [15] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, 2014, pp. 487–495.
- [16] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 633–641.
- [17] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
- [18] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kotschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4990–4999.
- [19] P. C. Pandey, N. Koutsias, G. P. Petropoulos, P. K. Srivastava, and E. Ben Dor, "Land use/land cover in view of earth observation: data sources, input dimensions, and classifiers—a review of the state of the art," *Geocarto International*, pp. 1–32, 2019.
- [20] D. Leung and S. Newsam, "Exploring geotagged images for land-use classification," in *Proceedings of the ACM Multimedia 2012 Workshop on Geotagging and Its Applications in Multimedia*, 2012, pp. 3–8.
- [21] Y. Zhu and S. Newsam, "Land use classification using convolutional neural networks applied to ground-level images," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2015, pp. 1–4.
- [22] V. Antoniou, C. C. Fonte, L. See, J. Estima, J. J. Arsanjani, F. Lupia, M. Minghini, G. Foody, and S. Fritz, "Investigating the feasibility of geo-tagged photographs as sources of land cover input data," *ISPRS International Journal of Geo-Information*, vol. 5, no. 5, p. 64, 2016.
- [23] L. Tracewski, L. Bastin, and C. C. Fonte, "Repurposing a deep learning network to filter and classify volunteered photographs for land cover and land use characterization," *Geo-spatial Information Science*, vol. 20, no. 3, pp. 252–268, 2017.
- [24] E. J. Hoffmann, M. Werner, and X. X. Zhu, "Building instance classification using social media images," in *2019 Joint Urban Remote Sensing Event*. IEEE, 2019, pp. 1–4.
- [25] S. Srivastava, J. E. Vargas-Muñoz, D. Swinkels, and D. Tuia, "Multilabel building functions classification from ground pictures using convolutional neural networks," in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, 2018, pp. 43–46.
- [26] Y. Movshovitz-Attias, Q. Yu, M. C. Stumpe, V. Shet, S. Arnoud, and L. Yatziv, "Ontological supervision for fine grained classification of street view storefronts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1693–1702.
- [27] T.-Y. Lin, S. Belongie, and J. Hays, "Cross-view image geolocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 891–898.
- [28] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5007–5015.
- [29] S. Workman, M. Zhai, D. J. Crandall, and N. Jacobs, "A unified model for near and remote sensing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2688–2697.
- [30] W. Zhang, W. Li, C. Zhang, D. M. Hanink, X. Li, and W. Wang, "Parcel-based urban land use classification in megacity using airborne lidar, high resolution orthoimagery, and google street view," *Computers, Environment and Urban Systems*, vol. 64, pp. 215–228, 2017.
- [31] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [32] M. Bar, "Visual objects in context," *Nature Reviews Neuroscience*, vol. 5, no. 8, pp. 617–629, 2004.
- [33] M. M. Chun and Y. Jiang, "Contextual cueing: Implicit learning and memory of visual context guides spatial attention," *Cognitive Psychology*, vol. 36, no. 1, pp. 28–71, 1998.
- [34] A. Torralba, K. P. Murphy, and W. T. Freeman, "Using the forest to see the trees: exploiting context for visual object detection and localization," *Communications of the ACM*, vol. 53, no. 3, pp. 107–114, 2010.
- [35] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 364–380.
- [36] Y.-F. Shih, Y.-M. Yeh, Y.-Y. Lin, M.-F. Weng, Y.-C. Lu, and Y.-Y. Chuang, "Deep co-occurrence feature learning for visual object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4123–4132.
- [37] C. Desai, D. Ramanan, and C. C. Fowlkes, "Discriminative models for multi-class object layout," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 1–12, 2011.

- [38] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.
- [39] M. J. Choi, A. Torralba, and A. S. Willsky, “Context models and out-of-context objects,” *Pattern Recognition Letters*, vol. 33, no. 7, pp. 853–862, 2012.
- [40] H. Izadinia, F. Sadeghi, and A. Farhadi, “Incorporating scene context and object layout into appearance modeling,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 232–239.
- [41] J.-T. Chien, C.-J. Chou, D.-J. Chen, and H.-T. Chen, “Detecting nonexistent pedestrians,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 182–189.
- [42] X. Wang, R. Girdhar, and A. Gupta, “Binge watching: Scaling affordance learning from sitcoms,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2596–2605.
- [43] X. Qiao, Q. Zheng, Y. Cao, and R. W. Lau, “Tell me where i am: Object-level scene context prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2633–2641.
- [44] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 1137–1149, 2017.
- [45] Z. Cai and N. Vasconcelos, “Cascade r-cnn: high quality object detection and instance segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [46] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
- [47] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder–decoder approaches,” in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014, pp. 103–111.
- [48] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [49] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, pp. 249–259, 2018.
- [50] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of freebies for training object detection neural networks,” *arXiv preprint arXiv:1902.04103*, 2019.
- [51] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [52] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [53] M. Grandini, E. Bagli, and G. Visani, “Metrics for multi-class classification: an overview,” *arXiv preprint arXiv:2008.05756*, 2020.
- [54] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, “Mmdetection: Open mmlab detection toolbox and benchmark,” *arXiv preprint arXiv:1906.07155*, 2019.
- [55] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.
- [56] J. Cheng, L. Dong, and M. Lapata, “Long short-term memory-networks for machine reading,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 551–561.
- [57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.