remote sensing

MDPI

Article

Model Fusion for Building Type Classification from Aerial and Street View Images

Eike Jens Hoffmann [1], Yuanyuan Wang [1], Martin Werner [2], Jian Kang [1] and Xiao Xiang Zhu [1,2,*]

[1] Signal Processing in Earth Observation, Technical University of Munich, 80333 Munich, Germany; eike.jens.hoffmann@tum.de (E.J.H.); y.wang@tum.de (Y.W.); jian.kang@tum.de (J.K.)

[2] Remote Sensing Technology Institute, German Aerospace Center, 82234 Weßling, Germany; Martin.Werner@dlr.de

* Correspondence: xiaoxiang.zhu@dlr.de

Received: 10 May 2019; Accepted: 22 May 2019; Published: 28 May 2019

Abstract: This article addresses the question of mapping building functions jointly using both aerial and street view images via deep learning techniques. One of the central challenges here is determining a data fusion strategy that can cope with heterogeneous image modalities. We demonstrate that geometric combinations of the features of such two types of images, especially in an early stage of the convolutional layers, often lead to a destructive effect due to the spatial misalignment of the features. Therefore, we address this problem through a decision-level fusion of a diverse ensemble of models trained from each image type independently. In this way, the significant differences in appearance of aerial and street view images are taken into account. Compared to the common multi-stream end-to-end fusion approaches proposed in the literature, we are able to increase the precision scores from 68% to 76%. Another challenge is that sophisticated classification schemes needed for real applications are highly overlapping and not very well defined without sharp boundaries. As a consequence, classification using machine learning becomes significantly harder. In this work, we choose a highly compact classification scheme with four classes, *commercial*, *residential*, *public*, and *industrial* because such a classification has a very high value to urban geography being correlated with socio-demographic parameters such as population density and income.

Keywords: street view image; aerial image; model fusion; building type classification; building function; CNN; urban land use; land cover

1. Introduction

Because of the past decade's rapid development of mobile devices, sensor technology, and particularly social media, we are now in an era with an immense number of optical images. These images comprise a wide diversity of modalities, from close-range photos taken with a smartphone, to spaceborne or aerial Earth observation images, and are acquired from distinctly different sensors and perspectives. They provide us a unique opportunity to understand the world better. The availability of such data has also inspired various applications, such as 3D reconstruction using ground-level and aerial images [1,2], localization using street view and aerial images [3,4], transformation between street view and satellite images [5], and joint classification using street view and satellite images [6,7]. This list is not exhaustive, but at the core of these applications lie two fundamental research questions and their concomitant challenges. One is the identification and extraction of street view and nadir view satellite/aerial images of the same location, and the other is the data fusion strategy that can cope with the different modalities of the two types of images. Thanks to the increasing number of geo-referenced ground-level images, such as those taken from smartphones, and those from map providers like Google Maps, the first task is becoming less challenging. Large datasets of street view and aerial image pairs such as CVUSA [8] are available, enabling the development of more sophisticated methods for addressing the actual data fusion problem. However, despite recent advances in computer vision and deep learning, this data fusion problem remains a challenge.

To this end, this article addresses the fusion of street view and nadir view satellite/aerial images via a generic building type classification task. We choose a classification scheme with four classes: *commercial*, *residential*, *public*, and *industrial*. The reason for this simplification is twofold. First, the building classes in sophisticated classification schemes for real applications are highly overlapping and not very well defined, which makes classification using machine learning significantly harder. Second, our classification scheme is highly valuable when using urban socio-demographic parameters such as population density and income, supporting the study of them and the development of future global products. Through our classification task, we will demonstrate the performance of different fusion strategies, including classification from individual image types, end-to-end two-stream convolutional neural networks (CNNs), and decision-level fusion by combining the predictions of different models.

Structure of This Article

The remainder of the article is structured as follows. The next section reviews the state of the art of land use land cover classification using ground view images, aerial view images, and both of them jointly. Section 3 introduces the dataset and the methods exploited in this article. Section 4 analyses the experimental results, and provides an explanation to new findings. Last but not least, Section 5 summarizes the most important findings of this article.

Throughout the paper, we use the vocabulary *network* or *CNN* to describe a network architecture, such as VGG; and use the vocabulary *model* to describe a trained network, or a fusion of many trained networks. One specific network may generate many models because it can be trained in different ways. We also use *street view*, *ground view*, and *ground-level* to describe terrestrial images that were taken from ground-level, and *aerial view*, *overhead*, and *nadir view* to describe remote sensing images acquired by airborne or spaceborne sensors.

2. Related Work

Urban land use classification has been a growing field of research as more image data have become available. These image data comprise both the ground view and the aerial view, but the different modalities have traditionally been investigated by different communities. The aerial view images have been mostly covered by the remote sensing community, while the ground view ones were mainly approached by the computer vision community. More detailed description of the state of the art to each aspect can be found in the following paragraphs in this section. A summary of the the state of the art is shown in Table 1.

2.1. Land Use Classification Using Aerial View Images

Earlier works on land use classification used handcrafted features extracted from remote sensing images. Hu and Wang extracted seven features from LiDAR and high-resolution images in their study area in Houston, Texas [9]. Using decision trees, they classified nine different parcel types with an overall accuracy of 61.68%. Random forests have been shown to be successful for urban land use mapping as well. By integrating spatial metrics and texture metrics, Hernandez and Shi reported an overall accuracy of 92.3% [10].

With the evolution of deep learning methods like CNNs, a shift from handcrafted to learned features was observed [11]. Marmanis et al. showed that an Overfeat network [12] pre-trained on ImageNet [13] and fine-tuned on the UC Merced Land Use dataset [14] achieves an overall accuracy of 92.4% on 21 classes [15]. Albert et al. explored the potential of two more recent architectures, VGG [16] and ResNet [17], on the Urban Atlas dataset [18]. They pre-trained on the DeepSat dataset [19], fine-tuned on the Urban Atlas dataset, and achieved an increase of about 5 percentage points in accuracy compared to pre-training on ImageNet. Their mean accuracy in six European cities is 50%

with ten different classes. Cheng et al. summarized three mainstream strategies for deep feature learning in remote sensing: (1) full training from scratch, (2) fine tuning, or (3) using CNNs only as feature extractors [20]. They conclude that "experimental results show that fine tuning tends to be the best performing strategy on small-scale datasets."

Beyond classification, Zheng et al. proposed a framework for semantic segmentation called OCNN [21] that relies on segmented objects as functional units instead of calculating pixel-wise convolution. Based on four bands, red, green, blue, and near infrared, they predicted ten land use classes in Southampton, UK, and nine classes in Manchester, UK. They reported 90.87% overall accuracy with a Kappa of 0.88 across both study areas.

2.2. Land Use Classification Using Ground View Images

Using ground view images for land use classification can be undertaken with photos from either social media platforms or map providers like Google Street View. Among the first to use social media images for land use classification were Leung and Newsam [22]. They downloaded images from Flickr to classify three types of buildings on two campuses: academic, sports, and residential. Using bag-of-words features derived from the images themselves as well as textual features from the image descriptions, they predicted a land use map with an SVM. With precision values up to 0.92, they showed that land use classification is feasible using social media images. Zhu and Newsam presented an improved approach by filtering images into two categories: indoor and outdoor [14]. Additionally, they replaced the bag-of-words features with features derived from a pre-trained network on the Places database [23] and achieved 76.84% accuracy on indoor images, compared to 80.85% accuracy on outdoor images. Kang et al. used Google Street View images to fine-tune several state-of-the-art CNN architectures for building instance classification [24]. To filter out images providing no information about the prediction class, they started by predicting all images obtained from the Google Street View API using a CNN trained on the Places database. Thus, images with occlusions like trees or vehicles or indoor scenes were left out for training. After filtering, all fine-tuning was performed on 17,600 images showing building facades across different cities in the US and labeled with eight building tags from OpenStreetMap (OSM). They reported an overall F1-score of 0.58 for a fine-tuned VGG16 CNN.

A similar approach was proposed by Srivastava et al. who used multiple Google Street View images of a building and fused them using a Siamese-like architecture [25]. Based on the VGG CNN model, they aggregated the fully connected layers by averaging. In their study area of Île-de-France, they collected 44,957 Google Street View pictures of 5941 OSM buildings. Predicting on 16 OSM labels, they achieved an overall accuracy of 62.52%. Since buildings in urban areas often have different usages at different floor levels, Srivastava et al. extended their approach to multilabel prediction [26]. With cadastral data of Amsterdam as the ground truth, they applied a CNN architecture using multiple images from Google Street View with varying fields-of-view to predict nine building function classes. By using three different fields-of-view, 30°, 60°, and 90°, they achieved an overall multilabel accuracy of 94.16%. Zhu et al. combined ground-view images from Google Places and Flickr to predict building instances [27]. By exploiting the multiple image categories both sources usually provide, they trained a two-stream CNN, where one stream uses Flickr images to predict objects and the other uses Google Places images to predict scenes. Additionally, they augmented their image dataset collected in San Francisco by searching for similar images using keywords in cities far from their study area (e.g., Paris, Atlanta, New York). Their hierarchical classification schema has 45 classes on the most fine-grained level. Using a fully trained ResNet101 CNN architecture, they showed 49.54% classification accuracy on image levels with 45 classes.

2.3. Land Use Classification Combining Ground and Aerial View

Combining both modalities was initially accomplished for image geo-localization. Lin et al. paired high-resolution satellite imagery from Bing together with ground-level images from Panoramio [28]. From both modalities, they extracted four handcrafted features and added land cover features as a third modality. By using these three modalities together, they were able to locate 17% of images coming from areas where there was no matching ground view image. They extended their approach by learning deep features between aerial and ground view images using pairs of Google Street View images in combination with bird's eye view images tilted 45 degrees downwards [29]. For this problem, the 45-degree view is necessary so that both images of a pair share some similarities. They showed that a 90-degree view combined with ground view is not suitable for finding a common representation. To fuse ground-view panoramas and 90-degree satellite images, Workman et al. proposed a unified model for near and remote sensing [6]. Using kernel regression, they integrate the ground-view images into a spatially dense feature map, which can then be used for fusion with the satellite image. Their network was trained end-to-end, including parameters for kernel regression. They used the resulting feature map for semantic segmentation applied to three different classification problems: land use, building function, and building age. In one of their two test datasets, Brooklyn, they report a top-1 accuracy of 77.40%, 44.88%, and 44.08% for land use, building function, and building age, respectively. Cao et al. used the same two datasets for land use classification with a two-stream encoder–decoder for semantic segmentation [7]. They extended the SegNet architecture [30] with a second encoder and fused each convolution layer with the first encoder network by stacking them together ahead of the max pooling layer. Their proposed fusion method achieved an overall accuracy of 78.10%, a Kappa coefficient of 73.10%, and an average F1-score of 62.73% for land use classification.

Table 1. Summary of different aspects of related work predicting land use with deep learning.

Aerial View	Ground View	Task	Dataset	Basic Architecture(s)	Method	Ref.
x		C	UC Merced Land Use	Overfeat	Fine-tuning from ImageNet	[15]
x		C	Google Maps satellite imagery	VGG, ResNet	Fine-tuning from ImageNet and DeepSat	[18]
x		S	High-res imagery from Manchester and Southampton	OCNN (based on AlexNet)	Markov process for joint learning two networks	[21]
	x	C	Flickr images from two university campuses	CaffeNet	Feature extraction from PlacesCNN and prediction with SVM	[22]
	x	C	Google Street View Imagery from 30 US cities	AlexNet, VGG, ResNet	Filtering with Places and then fine-tuning from ImageNet	[24]
	x	C	Google Street View imagery from Amsterdam	VGG	Finetuning from ImageNet and aggregating dense feature vectors using maximum or average	[25]
	x	C	Flickr and Google Street View imagery from San Francisco (augmented)	ResNet	Finetuning from ImageNet and Places to average probablity vectors	[27]
x	x	S	Bing aerial images and Google Street View from New York boroughs Queens and Brooklyn	VGG, PixelNet	End-to-end learning by stacking features and performing kernel regression on features	[6]
x	x	S	Bing aerial images and Google Street View from New York boroughs Queens and Brooklyn	SegNet	End-to-end learning by using a two stream encoder with a single stream decoder	[7]

C: classification. S: segmentation.

2.4. *Aspects of the Machine Learning Problem of Urban Land Use*

In contrast to many traditional remote sensing tasks, urban land use is highly complicated for a number of reasons. First of all, the applications of urban land use are interested in land use classes that are not measurable from space. Instead, they actually orient on the function of the building in the complex ecosystem of the city. In addition, there are instances where buildings have changed their function over time, for example, putting clubs or residential space into the manufacturing buildings of industries that have left the city. In addition, it is not clear how land use can be structured into a classification scheme at all. When defining classes from an application point of view, the classes will not be well-defined and will have significant overlap. For example, many buildings mainly serve residential purposes while still having shops and cafés inside. These issues need to be taken into account when designing the classification scheme.

2.5. Contribution of This Paper

Urban building type mapping has not been addressing using both remote sensing and street view images. This paper extends beyond state-of-the-art by exploiting two general aspects: first, the fact that the information contained in street view images and the information obtained from overhead imagery are different and can be combined to improved performance, and, second, the knowledge in huge collections of images in the datasets Places365 and ImageNet in order to understand the image content of both overhead imagery and street view scenery. To achieve these goals, a comprehensive comparison of existing models and fusion approaches was carried out. The contribution of this article is as follows:

- we compared two model fusion strategies: two-stream end-to-end fusion network (i.e., a geometric-level model fusion), and decision-level model fusion. Deep networks applied on individual data were also compared as baselines (i.e., no model fusion). A summary of the models and fusion strategies exploited in this article, as well as the corresponding literature, is shown in Table 2.
- we demonstrated that geometric combinations of the features of two types of images from distinct perspectives, especially combining the features in an early stage of the convolutional layers, will often lead to a destructive effect.
- without significantly altering the current network architecture, we propose to address this problem through decision-level fusion of a diverse ensemble of models pre-trained from convolutional neural networks. In this way, the significant differences in appearance of aerial and street view images are taken into account in contrast to many multi-stream end-to-end fusion approaches proposed in the literature.
- we have collected a diverse set of building images from 49 US states plus Washington D.C. and Puerto Rico. Each building in this dataset consists of a set of four images—one Google Street View image, and three Google aerial/satellite images at an increasing zoom level.

Table 2. Summary of the CNN models and different fusion strategies exploited in this article.

Basic Architecture(s)	Method	Section	Related Work
VGG, Inception	Fine-tuning from ImageNet	Section 3.2	[15,24]
VGG	Fine-tuning two stream network from ImageNet by stacking convolution layer horizontally	Section 3.3	[25,27]
VGG	Fine-tuning two stream network from ImageNet by stacking dense layer vertically	Section 3.3	[25,27]
VGG, Inception	Fine-tuning single stream from ImageNet and Places365 then blend decision layers	Section 3.4.1	[27]
VGG, Inception	Fine-tuning single stream from ImageNet and Places365 then stack decision layers with additional machine learning algorithm	Section 3.4.2	-

3. Methodology

In order to find the best fusion strategy, we performed a comparison of several state-of-the-art deep neural networks, as well as different model fusion strategies. A summary of the CNN architectures and fusion strategies exploited in this article, as well as the related literature, can be seen in Table 2.

3.1. The Datasets

In order to investigate fusion methods for building instance classification based on both remote sensing and street view images, a corresponding benchmark dataset was created for this paper. As illustrated in Figure 1, we extracted geolocation and the attributions of building function annotated by volunteers from OSM. Then, the associated street view images and the overhead remote sensing images of each building instance were retrieved via BingMap API and Google Street View API using its geolocation [31]. We set our program so that the retrieved street view images point toward the geolocation of each building. Three different zoom levels (17, 18, and 19 in the Google Maps convention) of overhead remote sensing images approximately centered at the building's geolocation were downloaded. The finest zoom level 19 is approximately 30 cm pixel spacing. The images cover 49 states (except Rhode Island) of the US, as well as Washington, DC and Puerto Rico, 51 areas in total. An example of our dataset can be seen in Figure 2, where the images of two buildings are displayed, one building per row. Despite the street view images point to each building, there is often occlusion due to existing vehicles and trees. This renders the fusion problem particularly challenging.

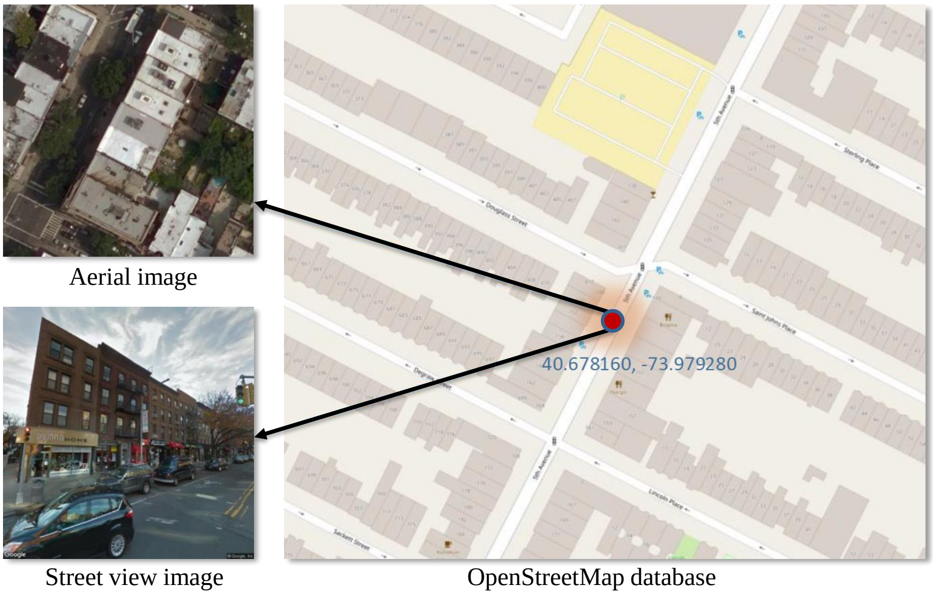


Figure 1. A illustration of the creation of the dataset. For each building, we look for the nearest Google street view image that pointing toward it, and the aerial image patch that centered on it. The label of the building is extracted from the OSM building tag.

Given the issues of urban land use described in Section 2.4, we follow a very basic but widely accepted classification scheme with four classes: *commercial*, *residential*, *public*, and *industrial*. To derive the class of each building, we extracted them from the volunteered building tag from OSM. However, as these tags are volunteered, their vocabulary can vary considerably, and even include spelling errors. Therefore, we selected the 16 most frequently occurring building tags in our raw dataset and aggregated them into four cluster classes: *commercial*, *industrial*, *public*, and *residential*. Table 3 shows the mapping and the number of buildings for each tag in detail. In summary, our dataset consists of 56,259 buildings with four images for each building. Among them, the images from the state of Wisconsin and Wyoming were used as validation samples (1943 buildings), those from the state of Washington and West Virginia were used as test samples (2212 buildings), and those from the remaining 47 areas were used as training samples (52,104 buildings).

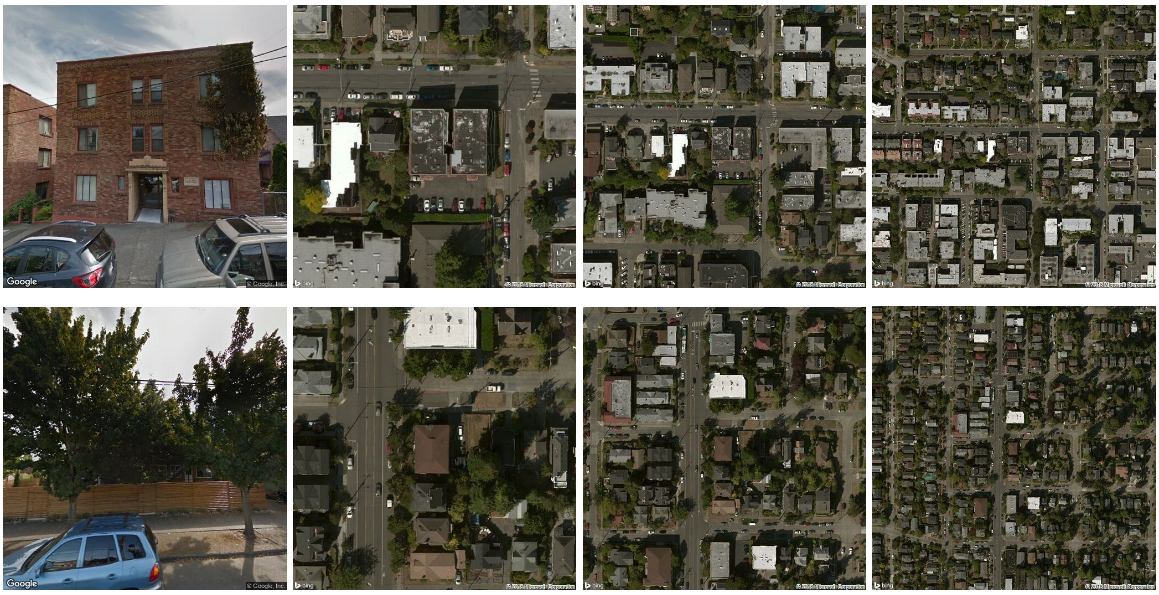


Figure 2. Examples of Google Street View and the corresponding overhead remote sensing images with zoom levels 19, 18, and 17 in our dataset. The street view image is pointing to the building instance. The remote sensing images are approximately centered on the building instance. As shown in the example on the second row, occlusion often happens in the street view images due to vehicles and trees. This renders the fusion problem particularly challenging.

Table 3. Mapping of OpenStreetMap building tags to general classes and instance numbers.

	Cluster Class	OpenStreetMap Tag	# of Buildings
1	commercial	commercial	5111
2	commercial	office	3306
3	commercial	retail	4906
4	industrial	industrial	3839
5	industrial	warehouse	2065
6	public	church	4153
7	public	college	1516
8	public	hospital	1758
9	public	hotel	2057
10	public	public	1966
11	public	school	4278
12	public	university	4020
13	residential	apartments	5039
14	residential	dormitory	2154
15	residential	house	5156
16	residential	residential	4935

It is important to note that, apart from the vocabulary difference and spelling error in the building tag, OSM also faces ambiguities in their finer classification scheme that is defined in the OSM Wiki. For many buildings, it is simply not possible to assign a single class. However, the OSM structure imposes the use of a single tag for each building; hence, the volunteer's choices significantly influence the consistency of the building tags. These inaccuracies of the OSM label in the training data and the simplification of the classification scheme are inevitable noise in the experiment set up. As a consequence, we cannot expect classifiers with 99% accuracy as is sometimes reported for land use classification in different contexts. Instead, a classification accuracy of about 60% to 80% on average would be a realistic expectation.

3.2. Fine-Tuning Exisiting CNNs for Individual Image Types

To obtain a baseline performance, we applied existing deep neural networks pre-trained on very large datasets, including Places365 [32] and ImageNet [33], on our street view and aerial images, respectively. These pre-trained models perform well for well-defined classification problems, where ImageNet is tailored to classes that refer to objects in images, while Places365 uses a classification scheme that already classifies street view scenery. However, we have to adapt these models for our classification scheme using an iterated fine-tuning approach, which is a standard approach in deep learning. First, we remove the softmax layer from the pretrained models and train for a few epochs with a new softmax layer fitting to the number of classes in our classification scheme. We apply a constant dropout to this layer such that only parts of the connections are available during training, while all connections will be used for inference. This technique is known to increase generalizability by forcing the neurons towards learning things that are universally useful rather than useful only in relation to other neurons [34]. When this final new layer has converged a little bit, we iterate by unlocking more layers and at the same time reducing the learning rate. In other words, we first train the last layer, than the last few layers, and so on. Finally, we take a very small learning rate and let the training continue with all layers. In this way, the network can gradually adapt to our case without destroying too much information in early layers due to fine-tuning with completely random final layers.

3.3. Fine-Tuning Two-Stream End-to-End Networks

The second method is two-stream end-to-end networks that is proposed in multiple literature for image fusion. We fine-tuned the existing CNN pre-trained on a large dataset for the two-stream network, using an approach similar to the training strategy described in the previous section. We selected VGG16 pre-trained on ImageNet as our base network in this section, as it will be demonstrated in Section 4.1 that different pre-trained networks provide comparable performance in the single-stream case. For the input data, we used the street view, and the aerial images with zoom level 19.

In the experiments, we fused the features of street view and remote sensing images in two slightly different methods: in one, we concatenated the bottleneck features (two 14*14*512 tensors) after the last convolutional layer of VGG16, and in the other we concatenated the features (two 4096*1 vectors) at the second to last dense layer of VGG16. The architectures of the two fusion models can be seen in Figure 3. In the first fusion model, we appended a convolutional layer with 64 filters, and three dense layers of 256 nodes each after the concatenated feature tensor. In the second fusion model, we simply concatenated the second to last dense layers of the two-stream VGG16, before the final dense layer. Batch normalization and dropout were also added after the concatenated features in both fusion models. The structure of the first fusion model, including the number of convolutional and dense layers, the number of filters in the convolutional layer, as well as the dropout rate, were determined using Bayesian optimization [35].

The fine-tuning consisted of two stages that were similar to the procedures described in the previous section. First, we lock the convolutional layers of VGG16 and use Bayesian optimization to select a relatively good set of hyperparameters, such as learning rate and dropout rate, for training the rest of the network. To reduce the computational effort, a maximum of 30 epochs was allowed for each trial in the Bayesian optimization. After 100 trials, the best set of hyperparameters was used to train the network for a dozen epochs, until there was a little bit more convergence. As mentioned above, the architecture of the first fusion model was also jointly optimized in this process. Afterwards, we progressively unlocked each convolutional block of the two-stream VGG16 network in three steps, at the same time reducing the learning rate by approximately one to two orders of magnitude.

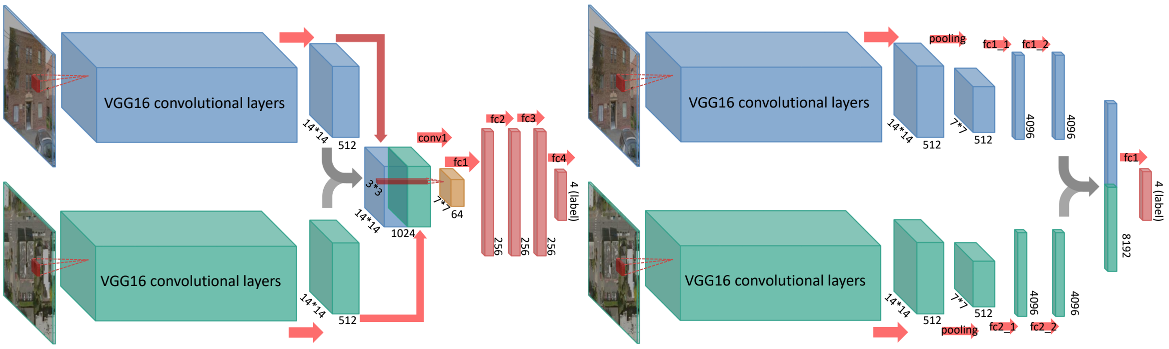


Figure 3. The two two-stream fusion models used in the article. The model on the left one concatenating the feature tensor (14*14*512) after the last convolutional layer of VGG16, and the right one concatenate the feature vector (4096*1) of the second last dense layer of VGG16. The fundamental difference is that the first model fuses the features earlier than the second model.

3.4. Decision-Level Model Fusion

Different from the two feature fusion strategies described in the previous section, decision-level fusion combines the softmax probabilities or the classification labels directly. We exploited two decision-level fusion strategies—*model blending* and *model stacking*—in this section. The architectures of the two fusion strategies are shown in Figure 4, where model blending takes the mean of the softmax layer of multiple models, while model stacking concatenates those softmax vectors. Both of the fusion strategies act on a decision level, which allows networks for individual data type to be trained independently.

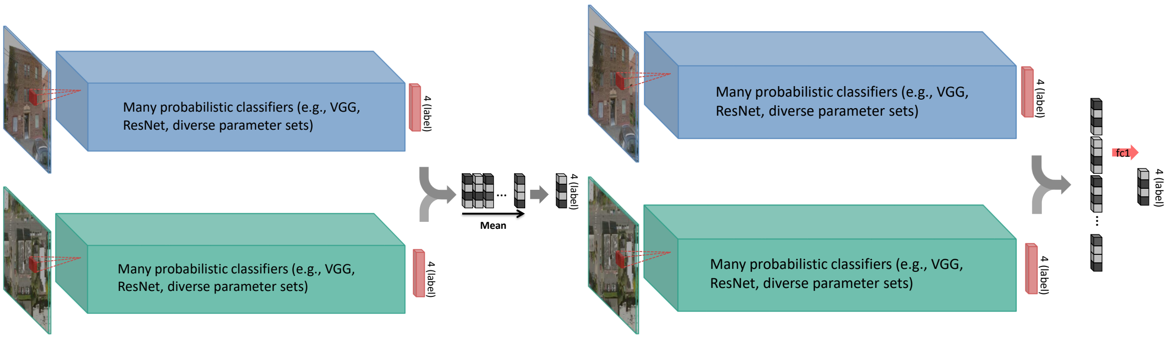

Many probabilistic classifiers (e.g., VGG, ResNet, diverse parameter sets)
4 (label)
Mean
fc1

Figure 4. A schematic drawing of the two decision-level fusion strategies—model blending (**left**) and model stacking (**right**)—exploited in this article. Model blending takes the mean of the softmax layer of multiple models, while model stacking concatenates those softmax vectors, and connects to a final softmax layer. Both of the fusion strategies act on a decision level, which allows networks for individual data type to be trained independently.

3.4.1. Fusion through Model Blending

The first decision-level fusion approach considered in our work is known as model blending. It is a very simple yet surprisingly powerful fusion scheme for probabilistic classifiers. In this case, the probability vectors of many different models are being averaged in order to create the probability vector that is then finally used for classification. In this way, if one modality is very certain about a class and the other modality is less certain, the average tends to select the right class from the certain model. If, however, both models are uncertain, it is likely that the average represents this uncertainty as well. In addition, it is possible that biases average out. As an extreme example, consider a model that chooses class A out of {A,B} with a probability vector of {0.6,0.4} and a second model B that does the opposite, i.e., chooses B with 0.6 and A with probability 0.4. Then, the average model will choose A and B with 0.5 probability. In this ideal example, two biased models have been blended and the bias is reduced. The downside of mean fusion is that it does not really allow for greedy strategies:

it is not clear that a model that is not performing very well is not able to clarify errors of another well-performing model. Therefore, all combinations of models must be checked for an exponentially growing number of

$$\frac{n!}{k!(n-k)!}$$

possible fusion models with n as the number of baseline models and k as the maximum number of models fused together. In order to deal with this situation, we first performed only pair-fusion with all our models and rejected models inside each modality that were outperformed by other models in all minimum, average, and maximum performance of any ensemble they were inside.

3.4.2. Fusion through Model Stacking

Another approach to the combination of machine learning models is generally known as *model stacking* and consists of using the individual models for feature extraction and then combining the resulting concatenated features into a single feature vector per item. In our case, we took the probabilistic vector output from each of the base models and concatenate them into a new vector—one vector for each building in the test and validation sets. Then, we could train a simple classifier on this vector, mapping the probabilistic outputs to the classes.

4. Experiments and Discussion

4.1. Performance of Existing CNNs on Individual Data

We performed the fine-tuning protocol described in Section 3.2 with varying numbers of parameters and base architectures on our datasets, and recorded the individual model performance given in Table 4. Without excessive tuning, we reached performances in the range of 57% precision (57% recall, F1 of 0.56) for one fine-tuned VGG-16 without global weight decay pretrained on Places365 to 68% precision (66% recall, F1 of 0.66) for an Inception model pretrained on ImageNet fine-tuned with aerial imagery of zoom level 19. The overall best model according to the Kappa score is a VGG-16 model pre-trained on ImageNet and fine-tuned with street view imagery. From this table, we can already see that the best two individual models are the best model from aerial and the best model from street view highlighting that both modalities are powerful for themselves.

Table 4. Performance of individual classifiers with varying modality and pre-training datasets.

Model	Precision	Recall	F1-Score	Kappa
Inception3-ImageNet-A19	0.68	0.66	0.66	0.52
Inception3-ImageNet-A18	0.68	0.63	0.60	0.47
Inception3-ImageNet-A17	0.59	0.55	0.51	0.33
VGG16-ImageNet-A19	0.65	0.60	0.59	0.42
VGG16-ImageNet-A18	0.60	0.60	0.59	0.42
VGG16-ImageNet-A17	0.59	0.55	0.54	0.34
Inception3-ImageNet-Streetview-1	0.63	0.58	0.58	0.41
Inception3-ImageNet-Streetview-2	0.63	0.59	0.59	0.42
Inception3-ImageNet-Streetview-3	0.62	0.57	0.58	0.40
Inception3-ImageNet-Streetview-4	0.62	0.57	0.57	0.39
VGG16-ImageNet-Streetview-1	0.67	0.67	0.67	0.53
VGG16-ImageNet-Streetview-2	0.61	0.59	0.59	0.41
VGG16-Places365-Streetview-1	0.57	0.57	0.56	0.37
VGG16-Places365-Streetview-2	0.67	0.65	0.65	0.49

Table 4 lists only a certain set of representative base models from several hundreds of models we have trained. For example, VGG16-Places365-Streetview-1 and VGG16-Places365-Streetview-2 differ only in the batch size (32 and 64, respectively). The general learning parameters are given in Table 5. In this table, *Batch* refers to the batch size used during stochastic gradient descent, *Decay* is the global weight decay parameter added to the error function, and N_i are the number of epochs that training is performed with learning rate l_i. The difference between the N_i is that we gradually unlock more layers during fine-tuning.

Table 5. Most important training parameters for the given models.

Model	Batch	Dropout	Decay	N_1	l_1	N_2	l_2	N_3	l_3
Inception3-ImageNet-A19	32	0.2	-	10	2×10^{-4}	10	2×10^{-4}	10	1×10^{-4}
Inception3-ImageNet-A18	32	0.2	-	10	2×10^{-4}	10	2×10^{-4}	10	1×10^{-4}
Inception3-ImageNet-A17	32	0.2	-	10	2×10^{-4}	10	2×10^{-4}	10	1×10^{-4}
VGG16-ImageNet-A19	32	0.2	1×10^{-5}	10	3×10^{-4}	50	3×10^{-4}	-	-
VGG16-ImageNet-A18	32	0.2	1×10^{-5}	10	3×10^{-4}	50	3×10^{-4}	-	-
VGG16-ImageNet-A17	32	0.2	1×10^{-5}	10	3×10^{-4}	50	3×10^{-4}	-	-
Inception3-ImageNet-Streetview-1	64	0.2	-	10	2×10^{-4}	10	2×10^{-4}	10	1×10^{-4}
Inception3-ImageNet-Streetview-2	32	0.2	-	10	1×10^{-4}	10	1×10^{-4}	10	5×10^{-5}
Inception3-ImageNet-Streetview-3	64	0.3	-	10	2×10^{-4}	10	1×10^{-4}	50	5×10^{-5}
Inception3-ImageNet-Streetview-4	64	0.35	-	10	5×10^{-5}	10	2×10^{-4}	20	1×10^{-4}
VGG16-ImageNet-Streetview-1	64	0.2	-	10	2×10^{-4}	10	2×10^{-4}	20	1×10^{-4}
VGG16-ImageNet-Streetview-2	32	0.2	1×10^{-4}	10	2×10^{-4}	10	2×10^{-4}	10	1×10^{-4}
VGG16-Places365-Streetview-1	32	0.2	1×10^{-4}	10	2×10^{-4}	10	2×10^{-4}	10	1×10^{-4}
VGG16-Places365-Streetview-2	64	0.2	1×10^{-4}	5	2×10^{-4}	10	1×10^{-4}	20	1×10^{-4}

In summary, we can conclude that individual modalities can be fine-tuned from pretrained weights into a performance range of about 50%–70% in all precision, recall, and F1 score.

4.2. Performance of Two-Stream End-to-End Networks

As mentioned in Section 3.3, the training of two-stream models consisted of two stages. In the first stage, we trained the networks with the convolutional layers of VGG locked, while, in the second stage, we progressively unlocked the VGG convolutional layers in a way similar to that described in Section 3.2. In the first stage of fine-tuning, the validation accuracy of both fusion models ranges from 40% to 60%. In the second stage, we attempted different combinations of decreasing learning rates, which are shown in Table 6. In this table, N_0 and l_0 refer to the number of epochs and the learning rate, respectively, of the best model in the Bayesian optimization step. N_1, N_2, N_3 and l_1, l_2, l_3 are the settings for the three steps in the second stage of fine-tuning. In the second stage of the fine-tuning, we found that a learning rate greater than 1×10^{-3} will not train the network at all. Therefore, we started with a learning rate of 1×10^{-4}.

Table 6. The most important training parameters for the two two-stream fusion models. N_0 and l_0 refer to the number of epochs and the learning rate of the best model in the Bayesian optimization step. N_1, N_2, N_3 and l_1, l_2, l_3 are the settings for the three steps in the second stage fine tuning, which are similar to those in Table 5.

Model	N_0	l_0	N_1	l_1	N_2	l_2	N_3	l_3
VGG16-Model1-1	20	1×10^{-5}	10	1×10^{-4}	20	1×10^{-5}	30	1×10^{-6}
VGG16-Model1-2	20	1×10^{-5}	10	1×10^{-4}	20	1×10^{-5}	30	1×10^{-7}
VGG16-Model1-3	20	1×10^{-5}	10	1×10^{-5}	20	1×10^{-6}	30	1×10^{-7}
VGG16-Model2-1	30	1×10^{-6}	5	1×10^{-4}	20	1×10^{-5}	30	1×10^{-6}
VGG16-Model2-2	30	1×10^{-6}	5	1×10^{-4}	20	1×10^{-5}	30	1×10^{-7}
VGG16-Model2-3	30	1×10^{-6}	5	1×10^{-5}	20	1×10^{-6}	30	1×10^{-7}

The performance of the fusion models evaluated on the test samples can be seen in Table 7. It is clear that the second fusion model outperforms the first one in general. The larger learning rate in the first step also helps to achieve better classification accuracy. However, compared to the model trained from individual image types in Section 3.2, the two fusion models do not show significant improvement in classification accuracy. Despite the second fusion model slightly outperforming the best individual VGG model in Table 4, the fusion often leads to a destructive effect. This is especially true for the first fusion model. We believe this is due to the misalignment of the geometry of the bottleneck features of the two image types. To illustrate this, an example of the bottleneck feature tensors (14*14*512) of an image pair in our dataset is shown in Figure 5, where the first, the 100th, and the 500th channel of the tensor are plotted. As we can see, the activated areas (in yellow and green) of the feature maps of the two images are distinct. A geometric fusion of those two feature maps, such as averaging or concatenation, will likely produce a destructive effect on the capability of pattern recognition. In contrast, the features after the dense layers of VGG16 contain less geometric information than the bottleneck features. Hence, better classification accuracy is achieved by fusing the feature vector after the dense layers. If further induction proceeds in a similar vein, it can be expected that the best performance will be achieved by a decision-level fusion of the output softmax probabilities of the two-stream network, which is basically training the two stream networks independently. Therefore, we decided to use a decision-level fusion of the models trained from individual data sources.

Table 7. The performance of the two fusion models w.r.t. different hyperparameters settings. The second fusion model in general outperforms the first one in general. Larger learning rate in the first step also helps to achieve better classification accuracy.

Model	Precision	Recall	F1-Score	Kappa
VGG16-Model1-1	0.63	0.62	0.62	0.50
VGG16-Model1-2	0.61	0.62	0.60	0.48
VGG16-Model1-3	0.66	0.61	0.61	0.47
VGG16-Model2-1	0.68	0.67	0.67	0.57
VGG16-Model2-2	0.66	0.67	0.66	0.55
VGG16-Model2-3	0.65	0.65	0.64	0.53

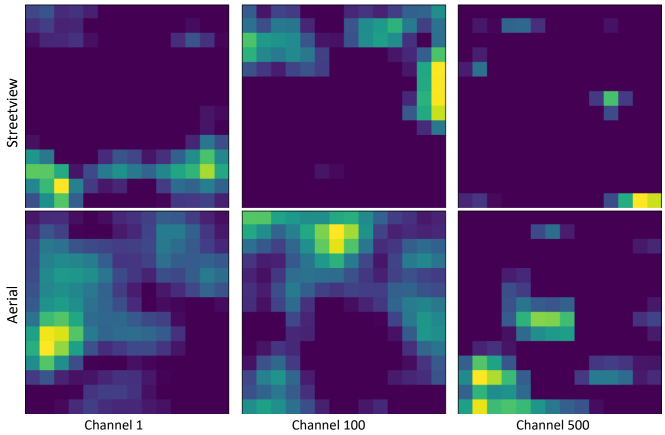
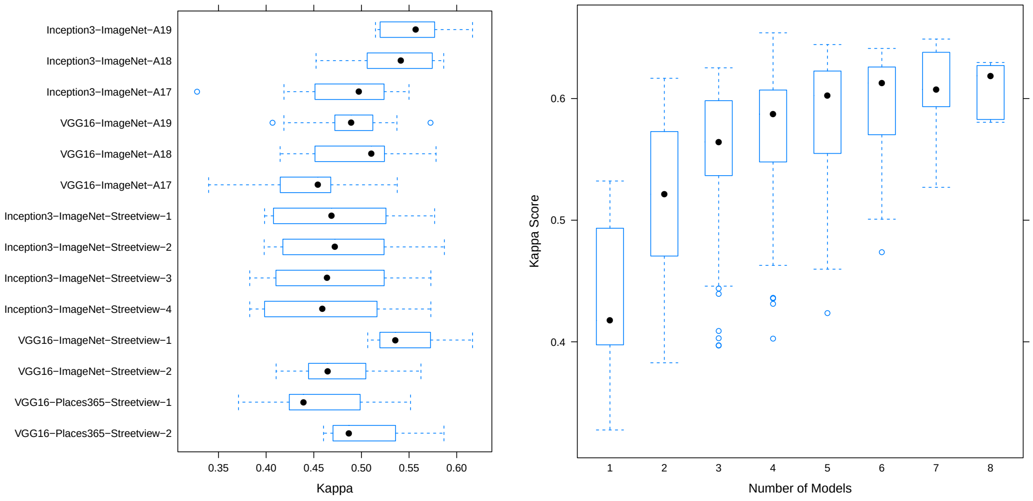


Figure 5. Example of the VGG16 bottleneck features of the street view (**upper row**) and the overhead remote sensing images (**lower row**) of one building in our dataset, which shows a probable reason of two-stream end-to-end fusion model *not* outperforming simpler decision-level fusion models. The first, 100th, and 500th channels of the bottleneck feature (14*14*512 tensor) of one image pairs in our dataset are plotted. We can see that the geometry of the feature maps in general do not align because a significant amount of spatial information is still contained in the bottleneck features. Such misalignment is common in most of the 512 bands as well as in most of the street view and aerial image pairs. A geometric fusion, such as average or concatenate, will likely produce a destructive effect on the capability of pattern recognition.

4.3. Performance of Decision-Level Fusion

4.3.1. Model Blending

Figure 6a depicts the statistics of the performance of ensembles which contain the specified model and exactly one other model. It clearly shows that ensembles containing aerial views fine-tuned from Inception outperform those fine-tuned using VGG16 architecture on average and on quantiles. Therefore, we do not include the VGG16-based models for the aerial layers in the final ensembling, expecting that their function is better fulfilled from Inception models for these modalities. Similarly, we remove VGG16-ImageNet-Streetview-2, as it is significantly outperformed by VGG16-ImageNet-Streetview-1 and we remove VGG16-Places365-Streetview-1, as it is outperformed by VGG16-Places365-Streetview-2 for the following complete subset fusion experiment on the remaining 10 models.



(**a**) (**b**)

Figure 6. Performance of ensembles with up to two members and performance of ensembles of varying size. The figure depicts a summary of the distribution of different ensembles formed with certain models inside or with certain numbers of models by indicating the distribution spread and mean. (**a**) Statistics of the set of Kappa values for all models built by fusing the named model with another model; (**b**) Performance relative to number of models.

For the remaining base models, we performed mean fusion and gave the best model results depending on the number of models we fused in Table 8 for up to four member models. Adding more models did not improve performance. Table 8 illustrates that the best model does contain the two extreme zoom levels as well as two different architectures for street view classification. The fusion process brings up the performance numbers from about 67% for the best individual model (cf. Table 4) to about 74%–76% precision and recall.

We analyzed the overall fusion approach and efficiency by looking into a selected set of base models and all possible fusion combinations out of this. The number of base models in this case was quite limited, as the number of possible subsets grows with the factorial of the number of base models. We then visualized two aspects of the overall fusion. First, we plotted the fusion model performance, given the number of base models in the fused model. This is depicted in Figure 6b. The figure clearly shows that the median performance, as given by the Kappa score, increases as the number of base models is added to the ensemble. In addition, the variance of the performance tends to decrease with the additional effect that the overall best model is not the model with the highest number of base models. Instead, it is one of the models with many, yet not too many models. In other words, while it is valid to expect the quality of models to increase by fusion, the largest model does not yield the best performance. Instead, the model with four elements discussed above is the overall best model from all possible fusions of the selected set of ten base models.

Table 8. Performance of the best mean fusion models with varying number of member models. It illustrates that the best model does contains the two extreme zoom levels as well as two different architectures for street view classification. The fusion process brings up the performance numbers from about 67% for the best individual model (cf. Table 4) to about 74%–76% precision and recall. Adding more models did not improve performance.

# of Models	Precision	Recall	F1-Score	Kappa	Models in Ensemble
2	0.74	0.73	0.73	0.62	Inception3-ImageNet-A19 VGG16-ImageNet-Streetview-1
3	0.74	0.74	0.73	0.63	Inception3-ImageNet-A18 VGG16-ImageNet-Streetview-1 Inception3-ImageNet-Streetview-2
4	0.76	0.76	0.75	0.65	Inception3-ImageNet-A19 Inception3-ImageNet-A17 VGG16-ImageNet-Streetview-1 Inception3-ImageNet-Streetview-4

The usefulness of the various base models was analyzed. For each model, Figure 7 shows a plot of the performance of all the blended models that contain the individual models listed in the figure. Note that in this case a large variance is actually a sign of a useful model: it has been used in bad models as well, which might just contain fewer element models. What is interesting about this plot is that our expectations can be clearly seen. For example, if the highly detailed zoom level 19 is part of an ensemble, then the overall ensemble tends to be better than if it only contains street view models. This fact can be derived because the difference in distributions does stem from all models that contain one but not the other as all models that contain both modalities come up in both distributions depicted in the figure. Consequently, we can see that street view has a significant contribution also independent from fusing it with zoom level 19 aerial imagery.

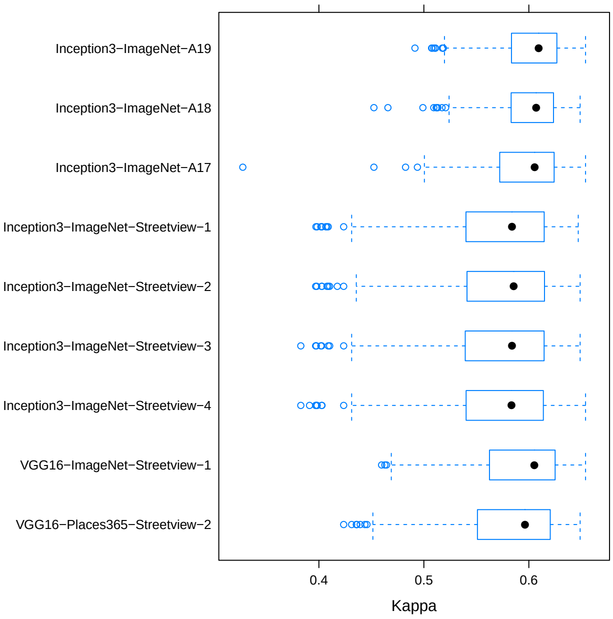


Figure 7. Performance of fusion models containing a selected model. The figure depicts a summary of the distribution of different ensembles formed with certain models inside or with certain numbers of models by indicating the distribution spread and mean. In the figure, our expectations can be clearly seen. For example, if the highly detailed zoom level 19 is part of an ensemble, then the overall ensemble tends to be better than if it contains street view models. In general, the performance of different model ensembles are comparable. In conclusion, we can see that both image modalities add independent value to the classification.

4.3.2. Model Stacking

In the previous section, we showed that mean fusion is already able to bring the individual multimodal models to a significantly improved fusion precision without investing any additional information, such as another train–test split. In the model stacking fusion strategy, we used the test set for training and the validation set for finally evaluating. In general, we have seen that this does not provide a significant improvement over the model blending from the previous section. We used logistic regression (75.2% precision, 73.2% recall), naive Bayes (72.9% precision, 71% recall), and Random Forests (75.1% precision, 72.3% recall). As can be seen, none of these models significantly outperforms the mean fusion performance.

Given that models that contain both aerial and streetview modalities will contribute identically to this figure, the variations come from models that use only one of the two mentioned modalities. In conclusion, we can see that both modalities add independent value to the classification. Still, these advanced stacking methods can be used to inject additional behavior into the classification that cannot be obtained from the base models that are trained on accuracy and cross entropy loss. For example, applying naive Bayes still leads to good values. What is particularly interesting is the fact that naive Bayes can work with minorities very well. This leads to a model with 56% precision for the industrial case, which is significantly higher than any of the other models. That is, for specific applications, the framework of stacking can well be used to steer into cost-sensitive classifications concentrating on certain classes. The results for this classifier (naive Bayes applied to the probabilistic output of all models on the test set, numbers extracted from the validation set) are depicted in Figure 8.

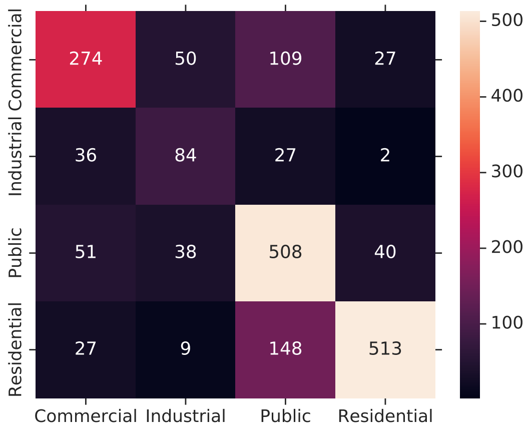


Figure 8. Performance of naive Bayes stacking for all models. This leads to a model with 56% precision for the industrial case, which is significantly higher than any of the other models. That is, for specific applications, the framework of stacking can well be used to steer into cost-sensitive classifications concentrating on certain classes.

In our situation, we think that the number of instances is too small to train significant classifiers on top of the output of the trained classifiers and that the effective reduction in available training data implied by additional train-test splitting turns down the effectiveness of this approach. Still, for significantly larger datasets, it is a promising direction because it can be more selective than model averaging. In fact, this approach could base decisions on data-varying subsets of classifiers, while the model blending case includes all classifiers into each decision. For our situation, this additional capability did not pay off. It is probable that a larger test set would improve the ability of learning in the stacking phase; however, it reduces the data for the individual model training. In essence, we conclude that, with the limited size datasets of remote sensing applications, mean fusion is the best approach as it does not consume additional data for training a second level of classifiers.

4.3.3. Influence of the Zoom Level on Classification Behavior

It is difficult to assess the value of each model in fusion settings, as it is always to be seen relative to the other models. If a model's individual performance is low, this does not mean that the model does not pay off in fusion. It could be, for example, very strong on the cases that some combination of other models is getting wrong and thereby could be adding a lot to the ensemble.

In order to still get some insight into the behavior of our classification problem with respect to aerial zoom levels, we analyzed the most simple models with different zoom levels in detail: we combined the best fine-tuned street view model with the best fine-tuned models for all selected zoom levels. Performances for these models are given in Table 9.

Table 9. Performance of fusion best street view model with best aerial models. It shows that higher-resolution imagery is more fruitful than the lower resolution counterparts in our setting. Interestingly, for the industrial class (not shown in the table), models of zoom 17 outperforms the rest. This is most likely related to the fact that some industrial buildings are very large and better represented in a lower zoom level.

Model	Precision	Recall	F1	Kappa
Streetview only	0.67	0.67	0.67	0.53
Streetview-Aerial 17	0.70	0.69	0.68	0.55
Streetview-Aerial 18	0.73	0.71	0.70	0.57
Streetview-Aerial 19	0.74	0.73	0.73	0.62

This table shows a clear trend for average performance: Higher-resolution imagery is more fruitful in our setting, as opposed to lower resolution imagery. However, when digging into the actual model details, we see another interesting aspect. For the industrial class, we get the following picture: the maximal F1-score is attained for both models with zoom 19 and with zoom 17; models with zoom 17 show a higher precision of 64% as opposed to 59% for zoom level 19. This is most likely related to the fact that some industrial buildings are very large and better represented in a lower zoom level. Similarly, the highest recall for the residential class is achieved from low-resolution imagery as well, with 93% recall. However, the precision in this case is comparably low, only 67%. With increasing resolution, the precision increases while the recall decreases (77% precision, 89% recall for zoom level 18; 84% precision with 80% recall for zoom level 19). In other words, when it comes to the classification of many buildings, the context given by larger zoom levels turns out to be very useful while at the same time increasing the probability of missing instances for a decreased recall.

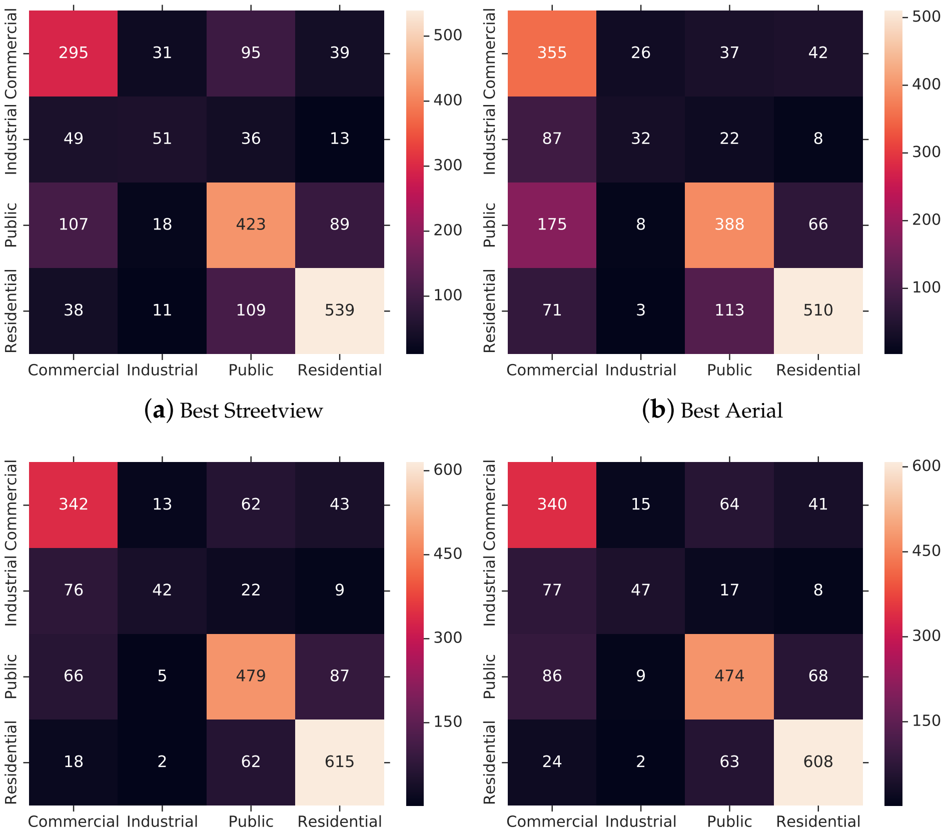
These findings are supported by the fact that the best fusion model among the chosen four models combines street view with zoom 17 and zoom 19, with performances of 74% precision, 74% recall, and 73% F1. In this case, zoom 18 is most likely left out because the information it can add is already part of the models for the neighboring zoom levels. In fact, the model fusion with all three zoom levels and street view is outperformed by four models taking into account only one or two of the aerial models.

Finally, the performance of any of the aerial models is lower than the performance of any fusion models that contain the street view perspective, and the ensemble models that contain only the street view models. This indicates that the street view perspective adds missing information to the usual remote sensing perspective and that the results of this paper could not be achieved from aerial observation only.

4.3.4. Best Model Discussion

Figure 9 depicts various confusion matrices for several best performing models. Several differences between the modalities are readily apparent. For example, the best street view model depicted in Figure 9a was able to correctly classify commercial buildings for only 64% of the cases, while the best aerial model depicted in Figure 9b reached 77% of the cases. This might be related to the fact that commercial buildings occur in patterns along major roads that are usually not visible in street view. On the contrary, public buildings are correctly classified in 66% of the case for street view as opposed to 61% for aerial. Looking into the actual misclassification, we see that this difference stems mainly from misclassifications into the commercial class. This is consistent with our intuition, as the settling structures visible from above should be quite similar for commercial and public buildings and their distinctions are easier from a street view perspective.

For the fusion models, we see that they outperform all single models in all four classes by a significant margin. It is interesting to observe, however, that the second-best fusion model is better with respect to the industrial class while worse with respect to the distinction of public and commercial classes. This is another hint that many of the top ensemble models can be relevant for application tasks and realize several trade-offs between classes.



(**a**) Best Streetview

(**b**) Best Aerial

(**c**) Best Fusion Model

(**d**) 2nd Best Fusion Model

Figure 9. Confusion matrices for four selected models. The best street view model depicted in subfigure (**a**) was able to correctly classify commercial buildings for only 64% of the cases, while the best aerial model depicted in (**b**) reached 77% of the cases. This might be related to the fact that commercial buildings occur in patterns along major roads that are usually not visible in street view. On the contrary, public buildings are correctly classified in 66% of the case for street view as opposed to 61% for aerial. Looking into the actual misclassification, we see that this difference stems mainly from misclassifications into the commercial class. From the confusion matrices of the fusion models in (**c**,**d**), we see that they outperform all single models in all four classes by a significant margin.

5. Conclusions

This article compared two different strategies—geometric feature fusion, and decision-level fusion—for fusing ground-level street view images and nadir-view remote sensing images with the application of building functions' classification. Our experiments conclude that, without sophisticated design of feature fusion mechanism in the network, a decision-level fusion of street view and overhead images often outperforms a feature-level fusion, despite its simplicity. Our explanation is that the misalignment of the geometry of features maps of the two image types will cause a destructive effect when combining them purely geometrically. This is especially true when combining the feature maps in an early stage of the convolutional layers. Therefore, this argument is also generally applicable to any images with distinct imaging perspective, geometry, or content, for example, radar and optical images.

To this end, we employed decision-level fusion strategies to achieve great performance without significantly altering the current network architecture. We let the individual networks for each image type be trained independently, so that the significant differences in appearance of aerial and street view images are taken into account, in contrast to many multi-stream end-to-end fusion approaches proposed in the literature. A significant performance boost can be further achieved by using a model ensemble, such as model blending and model stacking. Experiments showed that model blending without additional information, taking into account the uncertainty of the classifiers quantified in the softmax probabilistic layer, brings a significant gain. This approach brought classification precision from up to 68% for the best unimodal model to 76% for the best fusion model, taking into account street view and aerial imagery at the same time.

It is not surprising that the remote sensing images with the highest zoom level in general give better performance than those with less zoom level because of the higher spatial resolution. However, in the classification of residential areas, the image with the lowest zoom level outperforms the high-resolution images. This is because the contextual information helps to better determine residential buildings surrounded by similar ones. Therefore, our proposed method can be tailored to different applications, by combining different image types, zoom levels, as well as different models.

Author Contributions: Conceptualization, X.X.Z.; methodology, M.W., E.J.H., and Y.W.; software, M.W., E.J.H., and Y.W.; validation, M.W., E.J.H., and Y.W.; formal analysis, M.W., E.J.H., and Y.W.; investigation, M.W., E.J.H., and Y.W.; data collection, J.K., Y.W.; writing—original draft preparation, E.J.H., Y.W., M.W.; writing—review and editing, M.W., E.J.H., Y.W., and X.X.Z.; visualization, M.W., E.J.H., and Y.W.; supervision, X.X.Z.; project administration, X.X.Z.; funding acquisition, X.X.Z.

Funding: This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement number ERC-2016-StG-714087, Acronym: So2Sat, www.so2sat.eu), and Helmholtz Association under the framework of the Young Investigators Group "SiPEO" (VH-NG-1018, www.sipeo.bgu.tum.de).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application programming interface
CNN	Convolutional Neural Network
OSM	OpenStreetMap
SVM	Support Vector Machine
CVUSA	Cross-View USA dataset [8]

References

1. Koch, T.; Körner, M.; Fraundorfer, F. Automatic Alignment of Indoor and Outdoor Building Models Using 3D Line Segments. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 689–697. [CrossRef]
2. Rumpler, M.; Tscharf, A.; Mostegel, C.; Daftry, S.; Hoppe, C.; Prettenthaler, R.; Fraundorfer, F.; Mayer, G.; Bischof, H. Evaluations on multi-scale camera networks for precise and geo-accurate reconstructions from aerial and terrestrial images with user guidance. *Comput. Vis. Image Underst.* **2017**, *157*, 255–273. [CrossRef]

3. Bansal, M.; Sawhney, H.S.; Cheng, H.; Daniilidis, K. Geo-localization of street views with aerial image databases. In Proceedings of the 19th ACM International Conference on Multimedia—MM'11, Scottsdale, AZ, USA, 28 November–1 December 2011; ACM Press: New York, NY, USA, 2011; p. 1125. [CrossRef]
4. Majdik, A.L.; Albers-Schoenberg, Y.; Scaramuzza, D. MAV urban localization from Google street view data. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 3979–3986. [CrossRef]
5. Zhai, M.; Bessinger, Z.; Workman, S.; Jacobs, N. Predicting Ground-Level Scene Layout from Aerial Imagery. *arXiv* **2016**, arXiv:1612.02709.
6. Workman, S.; Zhai, M.; Crandall, D.J.; Jacobs, N. A Unified Model for Near and Remote Sensing. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
7. Cao, R.; Zhu, J.; Tu, W.; Li, Q.; Cao, J.; Liu, B.; Zhang, Q.; Qiu, G.; Cao, R.; Zhu, J.; et al. Integrating Aerial and Street View Images for Urban Land Use Classification. *Remote Sens.* **2018**, *10*, 1553. [CrossRef]
8. Workman, S.; Souvenir, R.; Jacobs, N. Wide-Area Image Geolocalization with Aerial Reference Imagery. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1–9.
9. Hu, S.; Wang, L. Automated urban land-use classification with remote sensing. *Int. J. Remote Sens.* **2013**, *34*, 790–803. [CrossRef]
10. Ruiz Hernandez, I.E.; Shi, W. A Random Forests classification method for urban land-use mapping integrating spatial metrics and texture analysis. *Int. J. Remote Sens.* **2018**, *39*, 1175–1198. [CrossRef]
11. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
12. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv* **2013**, arXiv:1312.6229.
13. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252.
14. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems—GIS'10, San Jose, CA, USA, 2–5 November 2010; ACM Press: New York, NY, USA, 2010; p. 270. [CrossRef]
15. Marmanis, D.; Datcu, M.; Esch, T.; Stilla, U. Deep Learning Earth Observation Classification Using ImageNet Pretrained Networks. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 105–109. [CrossRef]
16. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
18. Albert, A.; Kaur, J.; Gonzalez, M.C. Using Convolutional Networks and Satellite Imagery to Identify Patterns in Urban Environments at a Large Scale. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD'17, Halifax, NS, Canada, 13–17 August 2017; ACM Press: New York, NY, USA, 2017; pp. 1357–1366. [CrossRef]
19. Basu, S.; Ganguly, S.; Mukhopadhyay, S.; DiBiano, R.; Karki, M.; Nemani, R. DeepSat. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems—GIS'15, Seattle, WA, USA, 3–6 November 2015; ACM Press: New York, NY, USA, 2015; pp. 1–10. [CrossRef]
20. Cheng, G.; Han, J.; Lu, X. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]
21. Zhang, C.; Sargent, I.; Pan, X.; Li, H.; Gardiner, A.; Hare, J.; Atkinson, P.M. An object-based convolutional neural network (OCNN) for urban land use classification. *Remote Sens. Environ.* **2018**, *216*, 57–70. [CrossRef]
22. Leung, D.; Newsam, S. Exploring Geotagged images for land-use classification. In Proceedings of the ACM Multimedia 2012 Workshop on Geotagging and Its Applications in Multimedia—GeoMM'12, Nara, Japan, 29 October 2012; ACM Press: New York, NY, USA, 2012; p. 3. [CrossRef]

23. Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; Oliva, A. Learning Deep Features for Scene Recognition Using Places Database. In *Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 1*; MIT Press: Cambridge, MA, USA, 2014; pp. 487–495.
24. Kang, J.; Körner, M.; Wang, Y.; Taubenböck, H.; Zhu, X.X. Building instance classification using street view images. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 44–59. [CrossRef]
25. Srivastava, S.; Vargas-Muñoz, J.E.; Swinkels, D.; Tuia, D. 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, Seattle, WA, USA, 6 November 2018; pp. 43–46. [CrossRef]
26. Srivastava, S.; Vargas Muñoz, J.E.; Lobry, S.; Tuia, D. Fine-grained landuse characterization using ground-based pictures: A deep learning solution based on globally available data. *Int. J. Geogr. Inf. Sci.* **2018**, 1–20. [CrossRef]
27. Zhu, Y.; Deng, X.; Newsam, S. Fine-Grained Land Use Classification at the City Scale Using Ground-Level Images. *IEEE Trans. Multimed.* **2019**. [CrossRef]
28. Lin, T.Y.; Belongie, S.; Hays, J. Cross-View Image Geolocalization. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 891–898. [CrossRef]
29. Lin, T.Y.; Yin, C.; Belongie, S.; Hays, J. Learning deep representations for ground-to-aerial geolocalization. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5007–5015. [CrossRef]
30. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
31. Anguelov, D.; Dulong, C.; Filip, D.; Frueh, C.; Lafon, S.; Lyon, R.; Ogale, A.; Vincent, L.; Weaver, J. Google Street View: Capturing the World at Street Level. *Computer* **2010**, *43*, 32–38. [CrossRef]
32. Zhou, B.; Khosla, A.; Lapedriza, A.; Torralba, A.; Oliva, A. Places: An image database for deep scene understanding. *arXiv* **2016**, arXiv:1610.02055.
33. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the Conference on IEEE Computer Vision and Pattern Recognition, CVPR 2009, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
34. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
35. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2012; pp. 2951–2959.