BETTER GENERIC OBJECTS COUNTING WHEN ASKING QUESTIONS TO IMAGES: A MULTITASK APPROACH FOR REMOTE SENSING VISUAL QUESTION ANSWERING

Sylvain Lobry[1]*, Diego Marcos[1], Benjamin Kellenberger[1], Devis Tuia[1]

[1] Laboratory of Geo-Information Science and Remote Sensing, Wageningen University, The Netherlands - first.lastname@wur.nl

Commission II

KEY WORDS: Visual Question Answering, Regression, Deep learning, Remote sensing, Natural language, Convolution Neural Networks, Recurrent Neural Networks

ABSTRACT:

Visual Question Answering for Remote Sensing (RSVQA) aims at extracting information from remote sensing images through queries formulated in natural language. Since the answer to the query is also provided in natural language, the system is accessible to non-experts, and therefore dramatically increases the value of remote sensing images as a source of information, for example for journalism purposes or interactive land planning. Ideally, an RSVQA system should be able to provide an answer to questions that vary both in terms of topic (presence, localization, counting) and image content. However, aiming at such flexibility generates problems related to the variability of the possible answers. A striking example is counting, where the number of objects present in a remote sensing image can vary by multiple orders of magnitude, depending on both the scene and type of objects. This represents a challenge for traditional Visual Question Answering (VQA) methods, which either become intractable or result in an accuracy loss, as the number of possible answers has to be limited. To this end, we introduce a new model that jointly solves a classification problem (which is the most common approach in VQA) and a regression problem (to answer numerical questions more precisely). An evaluation of this method on the RSVQA dataset shows that this finer numerical output comes at the cost of a small loss of performance on non-numerical questions.

1. INTRODUCTION

Remote sensing can be used for a wide number of applications having direct impacts on people's life, including environment monitoring, urbanism, and land cover mapping. However, this source of information is hardly accessible to the majority of the population: While remote sensing data is easily available to scientists thanks to efforts such as ESA's Copernicus program, it is still a challenge to extract useful information from this data due to the technical nature of the task.

In terms of techniques, substantial efforts have been dedicated by the remote sensing community to automated information extraction from remotely sensed data, especially through deep learning models (Zhu et al., 2017; Reichstein et al., 2019).
Widely studied tasks include classification (Hu et al., 2015), semantic segmentation (Maggiori et al., 2017), regression (Yuan, Cheriyadat; Lobry et al., 2019), and object detection (Cheng, Han; Wu et al., 2019; Xia et al., 2018). Tasks are predominantly treated independently and addressed through dedicated models. However, a potential end-user might be interested in very specific information that those pre-defined models cannot deliver. For example, a citizen might want to know the number of playgrounds that have been built in the last five years and are less than 500 meters away from the house she is planning to buy. It is very unlikely for any existing off-the-shelf method to be able to answer such a question. This is mainly due to the fact that current models are built to solve a single specific task and therefore generalize poorly, also due to domain shifts (Tuia et al., 2016). In addition, they cannot process queries like a search engine would do, which further reduces versatility especially for non-experts. A potential solution to this issue has been proposed in Lobry et al. (2020), where the authors introduce Visual Question Answering (VQA) for remote sensing (RSVQA): VQA is a generic computer vision task that aims at providing an answer to an open-ended, free-form question about a given image (Antol et al., 2015).In VQA, both the question and the answer are expressed in natural language, lifting the technical aspect of formulating the task and communicating the model's output in an easily interpretable form for the layman.

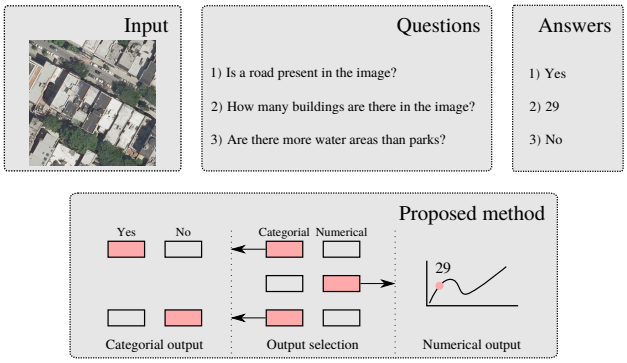

Figure 1. Schematic illustration of the proposed method.

VQA has been widely studied in the computer vision community (see section 2). Answering questions is most often cast as a classification problem, for which the goal is to predict the most probable answer among a set of pre-defined ones. However, this might not be suitable for some remote sensing tasks, where the output of questions of interest varies widely in its nature and might instead be more naturally connected to a regression or a semantic segmentation task. The overhead viewpoint further implies that numerical answers are often in a much

*Corresponding author

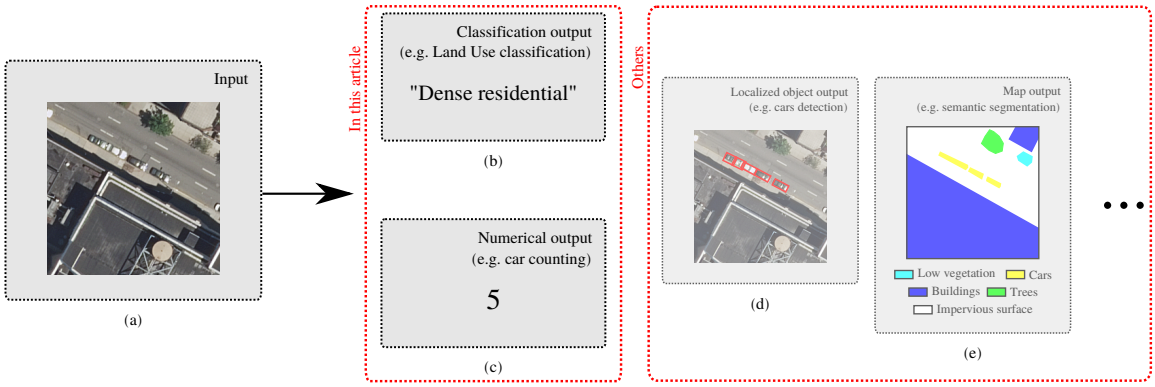

Figure 2. Examples of type of information which can be extracted from remote sensing images. (a) input image, (b) class of an image (in this case, the land use, (c) counts (in this case, cars), (d) localized objects (in this case, cars) and (e) map.

greater range than what could be found in natural images: for example, when counting buildings in a Sentinel-2 image over a densely populated city, one expects the answer to building counts to reach thousands, and this makes the formulation of the task as a classification problem impractical. In essence, remote sensing poses substantial complications to VQA models, both in terms of variability of answer types (classification, regression, semantic segmentation), and numerical ranges in the regression case in particular.

In this article, we tackle the problem of variability of tasks and answer ranges in RSVQA by adopting a multi-task approach (Caruana, 1997), illustrated in Figure 1. Our proposed model unifies the tasks of *explicitly recognizing the type of answer expected* and *providing the right format (e.g. class or number) of answer* in a single pipeline. We propose an RSVQA pipeline with three heads: the first analyzes the question, the second provides an answer in terms of answer classes (as in traditional VQA), and the third provides exact counts of objects for questions related to counting. By learning these three related objectives simultaneously, each task reinforces each other, leading to a more robust solution (Marmanis et al., 2018; Volpi et al., 2018).

We start with a review of the literature in section 2 and present a new multi-task architecture for VQA in section 3. The experiments are presented in section 4 and discussed in section 5.

2. RELATED WORK

2.1 Deep learning for Remote Sensing

A large number of works have been conducted using deep learning-based methods to extract information from remote sensing images. This information can be of different nature (*cf.* Figure 2), including:

1. **Classes** (Figure 2(b)): here, the objective is to assign a class (or category) to an image among a pre-defined set of classes, for instance land use (Yang, Newsam). In Penatti et al. (2015), the authors focused on the possibility to use pre-trained models to achieve this task. In details, a network is first trained on a large database (which can be from a different modality, *e.g.* ImageNet (Deng et al., 2009)) before being fine-tuned on the specific task. A similar study has been conducted by Hu et al. (2015), and Cheng et al. (2017), which both conducted an extensive evaluation of this kind of approaches for remote sensing image classification.

2. **Numerical** (Figure 2(c)): the prediction of numerical outputs from remote sensing can be useful for a wide range of topics, including environment monitoring, density estimation, or urban planning. For instance, Li et al. (2017) use a deep learning framework to first detect and then count palm oil trees. While a similar approach is used to count cars in Mundhenk et al. (2016), the problem can also directly be formulated as a regression task as in Lobry et al. (2019), where the objective is to count buildings.

3. **Localized objects** (Figure 2(d)): for this type of information, the problem (generally referred to as *object detection*) is to extract bounding boxes enclosing a particular object and to label them. An important effort was conducted by Xia et al. (2018), which localized 15 types of objects in large aerial images. This dataset enabled a number of studies: Li et al. (2019) generate region proposals with different orientations to account for the multiple orientation an object can have in remote sensing images.

4. **Maps** (Figure 2(e)): remote sensing images are georeferenced and cover most of the world and therefore can be used as resources for map creation. Several studies have tackled this problem with semantic segmentation approaches (Volpi et al., 2016; Maggiori et al., 2017) on the dataset released in the frame of the "semantic labeling contest" of ISPRS WG II/4[1].

2.2 Visual Question Answering

Despite the variety of information potentially available in remote sensing images, it remains a challenge to extract it due to the technical nature and specific focus of the methods presented above. In the computer vision community, the task of VQA has been introduced to enable natural interactions with images.

Among others, a topic of interest for recent works on VQA has been on counting questions. Indeed, it has been found to be one of the most challenging type of questions (Antol et al., 2015). Zhang et al. (2018) successfully used a specific counting module based on the interaction between object proposals and attention maps (*i.e.* a spatial weighting of the features based on the question). In Acharya et al. (2019), authors compiled counting questions from previously published VQA datasets, and created new, more complex questions involving reasoning. They also propose a model based on region proposals. In both

[1] http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html

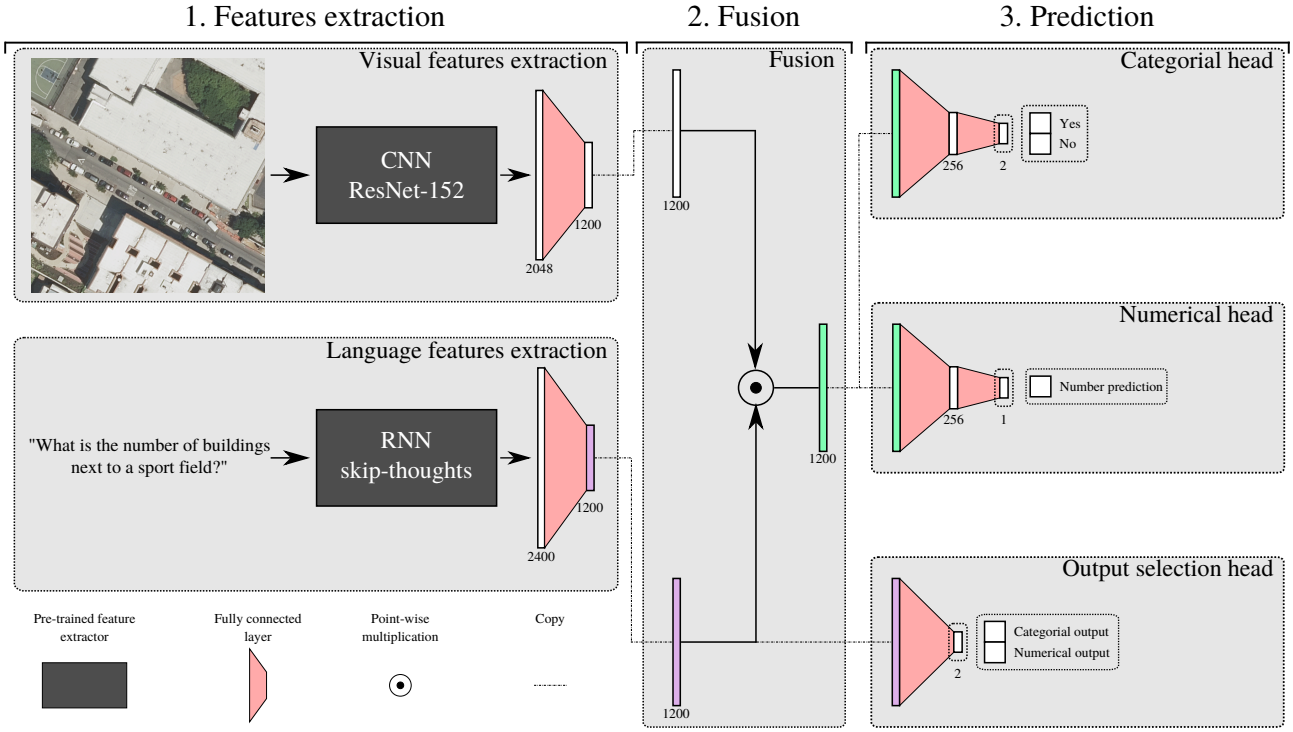

Figure 3. Architecture of the proposed model.

cases, these models consider the counting problem as a classification task, where the objective is to find the most probable answer among a set of pre-defined ones. This formulation becomes non-realistic when considering questions for which the answer could span several orders of magnitude. In Lobry et al. (2020), this was tackled by considering ranges (classes such as "the answer is between 1 and 10") rather than actual numbers. In this article we lift this constraint by formulating the counting problem as a regression task in a multi-task approach.

3. MODEL

The architecture of our proposed RSVQA model is presented in Figure 3. This model is based on three components (feature extraction, fusion and prediction), which are treated as a single optimization problem and learned end-to-end. While the architecture used to obtain the multi-modal feature vector (*i.e.*, feature extraction and fusion) is similar to the work presented in Antol et al. (2015), the prediction is formulated as a multi-task problem that involves both the prediction of the task type and the actual answering of the question.

Our goal is to predict an answer $\hat{\mathbf{y}}$ from a visual input $\mathbf{x_v}$ (a remotely sensed image) and a textual input $\mathbf{x_t}$ (a question). Note that the same image can be used to answer different questions and that the same question can be asked for multiple images, respectively. In our multi-task prediction framework, the answer can either be a textual answer (*e.g.*, "yes"), in which case we are solving a classification problem, or a numerical one (leading to a regression problem). To chose among the two possible outputs, we train a classifier predicting the type of answer to be expected from the question text (Figure 3, *output selection head*).

3.1 Feature extraction

The first component of our architecture extracts features from the visual and textual input. In both cases, we use the same strategy as in Lobry et al. (2020): a pre-trained feature extractor, followed by a mapping from the output of the model to a fixed sized vector (in our case, of dimension 1200).

Visual feature extraction To extract features from the remotely sensed image, we use a modified Resnet-152 model (He et al., 2016), which has been pre-trained on ImageNet (Deng et al., 2009). We replace the last average pooling layer and fully-connected layer with a per-pixel fully-connected layer that outputs a total of 2048 features for the image. This 2048-dimensional vectoris then mapped to 1200 dimensions using a single fully-connected layer.

Language feature extraction We obtain a feature vector representing the question using a recurrent neural network (skip-thoughts; Kiros et al. (2015)), pre-trained on the BookCorpus dataset (Zhu et al., 2015). This model has been trained on a sentence prediction task: it projects a sentence from a book in a latent space, and uses this representation to predict the previous and following sentences in the book. Using this method, the latent space encodes semantic information about the sentence. The output is a 2400-dimensional representation of our question. As for the visual part above, this feature vector is then projected to 1200 dimensions using a fully-connected layer.

3.2 Fusion

At this stage, we have two 1200-dimensional feature vectors, encoding the image and the question, respectively. These two vectors are merged using a simple point-wise multiplication to obtain a single 1200-dimensional feature vector.

3.3 Prediction

Our goal is to predict an answer $\hat{\mathbf{y}}$, which can either be textual or numerical. To this effect, and for each question/image input, our model predicts three outputs:

- A categorical answer $\hat{\mathbf{y}}_c$, from an N_c-dimensional vector whose elements are all possible admissible answers. Each dimension of the vector represents a score for the corresponding textual answer.
- A numerical answer $\hat{y}_n$, which is a scalar.
- An output selection $\hat{\mathbf{y}}_o$, which is a vector of dimension 2. Each dimension of this vector is a score indicating which type of answer to select as the actual prediction $\hat{\mathbf{y}}$ of our multi-task problem.

The categorical and numerical outputs are predicted through dedicated heads using a similar two-layer perceptron: first, the 1200-dimensional feature vector obtained at the fusion stage is projected to a 256-dimensional vector, which is itself projected to an n-dimensional vector (where n depends on the output, see Figure 3). Regarding the output selection branch, we use a single fully-connected layer that maps the 1200-dimensional textual feature vector from the language feature extractor to a vector of dimension 2.

3.4 Optimization

The proposed architecture is trained end-to-end in a supervised setting, using a combined loss with respect to the different outputs. We first extend the ground truth $\mathbf{y}$ in such a way that it is compatible with our prediction: from a ground truth that is a string representing the answer, we obtain a tuple as follows:

$$\mathbf{y} = (y_a, y_o) = \begin{cases} (y_c, 0) & \text{if the answer is categorical } (y_o = 0), \\ (y_n, 1) & \text{if the answer is numerical } (y_o = 1). \end{cases} \quad (1)$$

The loss is then defined as:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \mathcal{L}_o(\hat{\mathbf{y}}_o, y_o) + \alpha \mathcal{L}_c(\hat{\mathbf{y}}_c, \mathbf{y}) + \beta \mathcal{L}_n(\hat{y}_n, \mathbf{y})\,. \quad (2)$$

In Equation 2, $\mathcal{L}_o$ is a cross-entropy loss on the type of answer:

$$\mathcal{L}_o(\hat{\mathbf{y}}_o, y_o) = CE(\hat{\mathbf{y}}_o, y_o) = -\log\left(\frac{\exp(\hat{\mathbf{y}}_o^{y_o})}{\sum_i \exp(\hat{\mathbf{y}}_o^i)}\right)\,, \quad (3)$$

where $\hat{\mathbf{y}}_o^i$ is the i^{th} element of vector $\hat{\mathbf{y}}_o$. $\mathcal{L}_c$ is the cross-entropy conditional on y_o:

$$\mathcal{L}_c(\hat{\mathbf{y}}_c, \mathbf{y}) = \begin{cases} CE(\hat{\mathbf{y}}_c, y_c) & \text{if } y_o = 0\,, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

For the numerical answer, we use a mean-squared error on the logarithm of the ground truth, also conditional on y_o:

$$\mathcal{L}_n(\hat{y}_n, \mathbf{y}) = \begin{cases} w(y_n)\,(\hat{y}_n - \log(y_n + 1))^2 & \text{if } y_o = 1\,, \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

where $w(y_n)$ is a weight depending on the ground truth value (in our case, the inverse of the frequency of y_n in the training set). Note that we shift the logarithm function by 1 to account for the cases where $y_n = 0$. The prediction of our network will therefore be the logarithm of the desired answer (which can be retrieved as $\exp(\hat{y}_n) - 1$).

"What is the number of small buildings next to a road?"

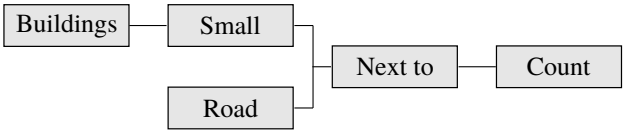

Figure 4. Sample question and corresponding tree representation.

We justify our choice for the MSE on the logarithm by the fact that numerical answers in the RSVQA task can span several orders of magnitude (*e.g.* from a single lake present in an image to many thousands of buildings over a densely built area). In this case, we argue that the precision of the answer is less critical than for low magnitude answers (*e.g.*, answering 2 when the answer should be 1 is more critical than answering 37902 instead of 37901). Following this intuition, the loss that we propose in Equation 5 penalizes wrong predictions based on the *ratio* with respect to the ground truth.

4. EXPERIMENTS AND RESULTS

4.1 Dataset

For our experiments, we use the high-resolution dataset introduced in Lobry et al. (2020)[2]. This dataset has been automatically derived from OpenStreetMap[3] (OSM) in the following way:

1. A question $\mathbf{x_t}$ about objects that are present in OSM layers (buildings, water areas, roads, *etc.*) is generated automatically. The question is first generated as a tree representation as shown in Figure 4, which is then converted into a sentence in English. Several types of questions are defined (*e.g.*, counting, presence, count comparison between two objects).
2. Using the OSM information corresponding to the extent of an image $\mathbf{x_v}$, the answer can also be retrieved automatically by parsing the tree representation using spatial queries in the OSM database.

The dataset generated following this procedure is composed of ($\mathbf{x_t}$, $\mathbf{x_v}$, $\mathbf{y}$) triplets on 10'659 tiles of 512×512 pixels and 15cm resolution from the USGS[4] acquired with various sensors. The tiles are located in New York and Philadelphia, USA, and have 1'066'316 questions about counts, comparison, presence or area of objects. Note that for this study, we discarded the questions related to the area of objects.

4.2 Training procedure

The models were trained using the Adam optimizer (Kingma et al., 2015), with a learning rate of 10^{-4}. The model is trained for 15 epochs. We use dropout (Srivastava et al., 2014) with probability 0.5 for every fully-connected layer and apply data augmentation (90° rotations and horizontal/vertical flipping) for questions that do not refer to positions relative to another object.

We experimentally fixed the parameters defined in Equation 2. For the experiments, α is set to 1 and β is set to 10.

[2]The dataset is available at `https://rsvqa.sylvainlobry.com`
[3]`http://www.openstreetmap.org`
[4]`https://earthexplorer.usgs.gov`

4.3 Metrics

We report the results using the accuracy (defined as the ratio of correct answers) in the case of classification tasks ($y_o = 0$). The average accuracy is computed by averaging the per-question type accuracies, and the overall accuracy is computed on the whole dataset of classification questions.

For counting tasks ($y_o = 1$), we report the Root Mean Squared Error (RMSE) as well as the Mean Absolute Error (MAE):

$$\text{RMSE} = \sqrt{\frac{1}{N_n} \sum_{(\hat{y}_n, \mathbf{y}) \text{ s.t. } y_o=1} \left((\exp(\hat{y}_n) - 1) - y_n\right)^2} \quad (6)$$

$$\text{MAE} = \frac{1}{N_n} \sum_{(\hat{y}_n, \mathbf{y}) \text{ s.t. } y_o=1} \left|(\exp(\hat{y}_n) - 1) - y_n\right| , \quad (7)$$

where N_n is the number of questions having a numerical answer. We also report the values computed on the logarithm of these metrics (LRMSE and LMAE) to account for the hypothesis made in subsection 3.4:

$$\text{LRMSE} = \sqrt{\frac{1}{N_n} \sum_{(\hat{y}_n, \mathbf{y}) \text{ s.t. } y_o=1} \left(\hat{y}_n - \log(y_n + 1)\right)^2} \quad (8)$$

$$\text{LMAE} = \frac{1}{N_n} \sum_{(\hat{y}_n, \mathbf{y}) \text{ s.t. } y_o=1} \left|\hat{y}_n - \log(y_n + 1)\right| \quad (9)$$

4.4 Results

We compare the results of our proposed model to the results of Lobry et al. (2020) (referred to as "Baseline"). Note that in the case of this dataset, counting was treated as a classification problem in the baseline, by including the possible numbers in the set of answers that can be predicted (as the maximum numerical answer was rather low, 86).

The classification and output selection scores are presented in Table 1 and the regression scores in Table 2. A scatter plot of the regression results of the baseline and the proposed model is shown in Figure 5. Finally, some visual examples are shown in Figure 6.

Table 1. Classification results on the test set of the high resolution dataset.

Type	Baseline	Proposed model
Presence	90.43%	89.43%
Comparison	88.19%	84.40%
AA	89.28%	86.91%
OA	89.14%	86.64%
Output selection	N.A.	100%

Table 2. Counting results on the test set of the high resolution dataset.

Type	Baseline	Proposed model
RMSE	2.57	2.82
MAE	0.94	1.11
LRMSE	0.44	0.40
LMAE	0.20	0.27

5. DISCUSSION

Numerical prediction performances:
When comparing the regression results over the entirety of the dataset (presented in Table 2), the proposed model slightly underperforms compared to the baseline. However, note that this is not equally distributed among the different possible answers (Table 3), as the proposed model performs better in the range of answers between 1 and 10. Several reasons could explain this non-uniformity:

Table 3. Per-range counting results on the test set of the high resolution dataset. Bold indicate the best result for each metric and each range.

Type	Baseline			Proposed model		
Range	0	1-10	11-54	0	1-10	11-54
RMSE	0.63	2.32	**8.48**	**0.62**	**2.19**	9.80
MAE	**0.07**	1.49	**6.37**	0.22	**1.48**	7.61
LRMSE	**0.20**	0.62	0.83	0.27	**0.50**	**0.70**
LMAE	**0.04**	0.43	**0.52**	0.16	**0.40**	0.54

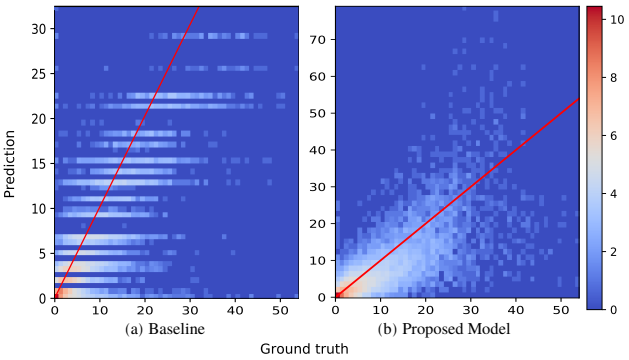

Figure 5. Scatter plots (in log scale) of the counting prediction for the (a) baseline and (b) proposed model. The red lines indicate a theoretical perfect predictions.

- **Case** $y_n = 0$: this case is the vast majority of the database, with 60.6% of the numerical answers being 0 in the test set. In this case, we have a great imbalance—this is an advantage for the baseline, which did not use class weights in the cross-entropy loss. At the same time, it is an edge case in the sense that our model almost always over-predicts in this range (since it tends to predict only positive numbers).
- **Case** $1 \leq y_n \leq 10$: in this range, our proposed model performs better, likely thanks to the loss defined in Equation 5. This loss tends to favor good performances for relatively low numbers, as it penalizes the ratio between the prediction and the ground truth, rather than the difference. This is confirmed in Table 3, where it is the only range in which the proposed model consistently performs better over the four metrics.
- **Case** $10 > y_n$: For larger numbers, wrong predictions are less penalized by the logarithm-based loss. However, this range only account for 6.4% of the test set, which validates the hypothesis made for the use of the logarithm-based loss.

Besides numerical evaluation, it is important to look at the scatter plots shown in Figure 5. While the spread around the perfect prediction line (red) is larger for the proposed model, also the baseline suffers from a number of issues, mainly due to the framing of the problem as classification. First, the baseline never predicts numbers larger than 33, due to the strong imbalance in the dataset. Second, the predictions are clustered around a few possible values, and also some numbers in-between are

(a) (b) (c)

Figure 6. Predictions of the proposed model on the test set.

never predicted. Both of these problems do not appear in the proposed model.

When looking at some predictions of the proposed model, we can see in Figure 6(a) that it is able to correctly answer complex questions involving spatial reasoning. On the opposite, it seems that it is more complicated to understand the arrangement of buildings, as highlighted in Figure 6(b). Finally, we show in Figure 6(c) an example of an outlier in our database: while the prediction is reasonable, the fact that this area is not up-to-date in the OSM database gives a wrong ground truth for this sample. The presence of outliers particularly affects the regression losses, as studied in Lobry et al. (2019). This could explain the lower scores of our proposed model, despite the smoother predictions shown in the scatter plot.

Impact on the other tasks:
When looking at the non-numerical tasks, we can notice a small performance decrease: -1% on the presence task, -4% on the comparison task. This loss of performance can be attributed to the multi-task aspect of our proposed model, reflected in Equation 2: for the baseline, a single cross-entropy loss was used, and the answers to classification questions are present in large quantities compared to most numbers (except for 0). Therefore, the cross-entropy loss, being sensitive to imbalance, encourages the learning process to focus on these tasks. On the contrary, the multi-task loss of Equation 2 makes sure that an important effort is dedicated to counting, leading to a slight performance decrease on other tasks.

Multi-task framework:
We can see in Table 1 that the model achieves a 100% accuracy for the output selection. This validates that the multi-task formulation proposed in this work can be efficient, as the task identification is easy in this case. Indeed, the type of answer can be derived solely from the presence of certain keywords: *e.g.* "How many", "What is the number", *etc.* will always have a numerical answer. This figure would change on a dataset with human-made questions as the automatic generation used in the RSVQA dataset is very simple.

Size of the network:
The proposed model has a similar size as to the baseline (0.3% increase of the total number of parameters). This is due to the fact that the baseline has a larger final fully-connected layer since it needs to take into account possible numbers. Therefore, the proposed model does not add a significant complexity overhead.

6. CONCLUSION

We introduced a multi-task model for RSVQA, which predicts answers of different nature (categorical and numerical) and returns the most relevant one of the two depending on its interpretation of the question. Analyses show that the multi-task formulation is relevant for RSVQA, with the proposed model providing more sensible and fine-grained values for regression-related questions. This covers the requirement for precise regression on the one hand, and addresses problems related to large numerical ranges on the other. However, the prediction of numerical answers remains a difficult problem, as previously found in research from both the remote sensing and computer vision communities. Future work could be dedicated to improving the numerical prediction by taking into account oultliers either through a robust loss, or by regularizing the network (Damodaran et al., 2019). Furthermore, adding different types of information (*e.g.* localized objects, maps) to the possible outputs could further extend the expressivity of RSVQA methods, making them even more approachable to non-experts in the field.

ACKNOWLEDGEMENTS

The authors would like to thank CNES for the funding of this study under the R&T project "Application des techniques de Visual Question Answering à des données d'imagerie satellitaire".

References

Acharya, M., Kafle, K., Kanan, C., 2019. TallyQA: Answering complex counting questions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 8076–8084.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., Parikh, D., 2015. VQA: Visual Question Answering. *International Conference on Computer Vision*, 2425–2433.

Caruana, R., 1997. Multitask learning. *Mach. Learn.*, 28, 41-75.

Cheng, G., Han, J., 2016. A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117, 11–28.

Cheng, G., Han, J., Lu, X., 2017. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10), 1865–1883.

Damodaran, B. B., Fatras, K., Lobry, S., Flamary, R., Tuia, D., Courty, N., 2019. Wasserstein Adversarial Regularization (WAR) on label noise. *CoRR*, abs/1904.03936. http://arxiv.org/abs/1904.03936.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hu, F., Xia, G.-S., Hu, J., Zhang, L., 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11), 14680–14707.

Kingma, D., Ba, J., ., 2015. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R., Torralba, A., Urtasun, R., Fidler, S., 2015. Skip-thought vectors. *Neural Information Processing Systems*, 3294–3302.

Li, Q., Mou, L., Xu, Q., Zhang, Y., Zhu, X. X., 2019. R^3-Net: A deep network for multioriented vehicle detection in aerial images and videos. *IEEE Transactions on Geoscience and Remote Sensing*, 57(7), 5028–5042.

Li, W., Fu, H., Yu, L., Cracknell, A., 2017. Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sensing*, 9(1), 22.

Lobry, S., Marcos, D., Murray, J., Tuia, D., 2020. RSVQA: visual question answering for remote sensing data. *submitted to IEEE Transactions on Geoscience and Remote Sensing*.

Lobry, S., Tuia, D., ., 2019. Deep Learning Models to Count Buildings in High-Resolution Overhead Images. *Joint Urban Remote Sensing Event*.

Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., 2017. High-resolution aerial image labeling with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12), 7092-7103.

Marmanis, D., Schindler, K., Wegner, J. D., Galliani, S., Datcu, M., Stilla, U., 2018. Classification with an edge: improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135, 158 - 172.

Mundhenk, T. N., Konjevod, G., Sakla, W. A., Boakye, K., 2016. A large contextual dataset for classification, detection and counting of cars with deep learning. *European Conference on Computer Vision*, 785–800.

Penatti, O. A., Nogueira, K., Dos Santos, J. A., 2015. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? *IEEE conference on computer vision and pattern recognition workshops*, 44–51.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., 2019. Deep learning and process understanding for data-driven Earth system science. *Nature*, 566, 195-204.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929–1958.

Tuia , D., Persello, C., Bruzzone, L., 2016. Recent advances in domain adaptation for the classification of remote sensing data. *IEEE Geoscience and Remote Sensing Magazine*, 4(2), 41-57.

Volpi, M., Tuia, D., ., 2016. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), 881–893.

Volpi, M., Tuia, D., ., 2018. Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 144, 48-60.

Wu, X., Hong, D., Tian, J., Chanussot, J., Li, W., Tao, R., 2019. ORSIm Detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features. *IEEE Transactions on Geoscience and Remote Sensing*, 57(7), 5146–5158.

Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L., 2018. DOTA: A large-scale dataset for object detection in aerial images. *IEEE Conference on Computer Vision and Pattern Recognition*, 3974–3983.

Yang, Y., Newsam, Shawn, ., 2010. Bag-of-visual-words and spatial extensions for land-use classification. *SIGSPATIAL international conference on advances in geographic information systems*, 270–279.

Yuan, J., Cheriyadat, A. M., 2014. Learning to count buildings in diverse aerial scenes. *SIGSPATIAL international conference on advances in geographic information systems*, 271–280.

Zhang, Y., Hare, J., Prügel-Bennett, A., 2018. Learning to count objects in natural images for visual question answering. *International Conference on Learning Representations*.

Zhu, X., Tuia, D., Mou, L., Xia, G., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4), 8-36.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S., 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *IEEE international conference on computer vision*, 19–27.