

# Weakly Supervised Fusion of Multiple Overhead Images

Muhammad Usman Rafique Hunter Blanton Nathan Jacobs  
University of Kentucky

{usman.rafiq, hunter.blanton, nathan.jacobs}@uky.edu

## Abstract

This work addresses the problem of combining noisy overhead images to make a single high-quality image of a region. Existing fusion methods rely on supervised learning, which requires image quality annotations, or ad hoc criteria, which do not generalize well. We formulate a weakly supervised method, which learns to predict image quality at the pixel-level by optimizing for semantic segmentation. This means our method only requires semantic segmentation labels, not explicit artifact annotations in the input images. We evaluate our method under varying levels of occlusions and clouds. Experimental results show that our method is significantly better than a baseline fusion approach and nearly as good as the ideal case, a single noise-free image.

## 1. Introduction

In recent years, it has become possible to frequently collect high resolution overhead imagery of the same location. Given the nature of how the images are captured and registered, they often contain artifacts such as clouds, large streaks of missing data, significantly varying lighting and weather condition, and other artifacts [6]. These are major issues when attempting to use these images for remote sensing tasks, such as object detection or semantic segmentation.

Several methods have been proposed that address the problem of such artifacts. Generally these methods require either problem-specific solutions or expensive labeling of artifacts in input images. While the former techniques do not require labelled data, the solutions typically require several complex interpolations and image processing techniques. These methods generalize poorly and require re-calibration to newer scenarios. On the other hand, learning based methods have shown promising results in detecting artifacts like clouds and cloud shadows. However, these methods require labelled training data with every pixel categorized as clean, partly occluded, and fully occluded. Understandably, it is an expensive and subjective process to

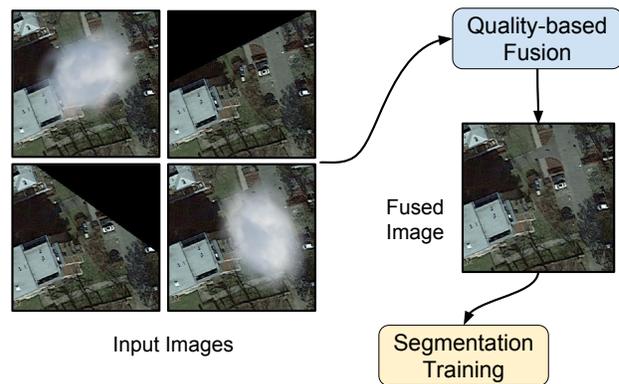


Figure 1: Overview of the proposed approach. We present a fusion method which can fuse multiple images of a region: these images might have artifacts like missing data or occlusion by clouds. Our method is weakly supervised as we do not need labels for missing data or clouds, we just need dense labels for semantic segmentation.

label pixels accurately. Recently, some convolutional neural networks (CNNs) have shown invariance to clouds while working on multiple images for the task of vegetation classification. Alternatively, given the large number of images captured at the same location, it is possible to fuse information of several images into a single image. This single image can then be used for a variety of tasks.

In this work, we explore the problem of fusing multiple registered overhead images. The goal is for the fused image to produce more accurate predictions for the task than the individual input images. This is performed using a per-pixel scoring CNN, which we call a quality network, that predicts quality for each image pixel. A high quality pixel is one that is more useful for the task. In this work, we consider semantic segmentation as the final task. Based on scores from the quality network, we synthesize the fused image directly from the input images.

We present a general framework which learns to combine multiple input images, based on the quality of every

pixel, such that the performance of the final task of semantic segmentation is improved. An overview of our method is shown in Figure 1. We pass all input images through a quality network that scores each pixel. The final image is synthesized based on these scores. The fused image is then fed to a standard image segmentation network. This allows us to train an end-to-end system that learns to predict per-pixel quality without requiring labels for artifacts in individual images. We only need semantic segmentation labels, such as roads and buildings, to optimize our method. We quantitatively and qualitatively show that our system learns to identify image artifacts to make a final image that is cleaner than any of the individual input image. Since we synthesize a single fused image which is better than the noisy input images, it can then be used for any task.

Our main contribution is that we propose a weakly supervised learning approach that learns to predict image quality without requiring image quality labels. To the best of our knowledge, this is first work that both synthesizes clean images, without direct supervision, and optimizes for a task like semantic segmentation using multiple images. We systematically show the usefulness of our method in various situations with varying number of input images and amount of artifacts in images.

## 2. Related Work

The problem of fusing multiple overhead images of a region has been explored in various classical and modern methods. These methods require different levels of input information and the solutions range from detection of artifacts to fusion of multiple noisy image.

### 2.1. Classical Fusion Methods

A classical, Fourier analysis based interpolation method was presented by Roerink *et al.* [13]. In this method, missing and cloudy regions were filled in by interpolating images collected at other times. Results were shown on Normalized Difference Vegetation Index (NVDI) composite images of agricultural areas in Europe. The method assumes knowledge of frequencies in images based on fast Fourier transform of previous, presumably clean, imagery. A patch based method was presented by Lin *et al.* [10]. This method included several steps including cloud detection, intensity normalization, and patch-based image synthesis to remove clouds from multiple images. It is a complex method and some steps are slow, e.g. seam selection is formulated as an optimization problem. A similar method by Chen *et al.* [1] proposed a method to merge target and reference images to remove clouds from given images. It is assumed that clouds can be perfectly detected using Fmask [20], a classical detection method for Landsat imagery.

### 2.2. Cloud Detection

A key component of most existing methods is a separate module for cloud detection. Note that these methods rely on explicit labels of clouds, either available at test time or through some existing algorithm. Several datasets have been prepared to evaluate methods of cloud detection, for example Landsat-8 SPARCS by Foga *et al.* [3]. In this dataset, pixel-wise dense labels are provided that include cloud, shadow or clean. A supervised cloud detection method was presented by Li *et al.* [9]. This method requires labels of clouds and cloud shadows. Once trained, their method can label every pixel as clean, cloudy, or having cloud shadow: they do not try to make a clean image using these masks.

### 2.3. Removal of Translucent Clouds

Several methods, ranging from classical image processing to modern generative adversarial networks (GANs), have been proposed to remove translucent clouds from overhead imagery. Enomoto *et al.* [2] presented a method of removing thin clouds from aerial images. They built a synthetic dataset in which Perlin noise is used to simulate clouds. They used a multispectral conditional GAN to remove clouds from images. A major limitation of their method is that the near IR image is not covered with clouds. Singh and Komodakis [18] used a conditional GAN on color images. This paper also used Sentinel-2 imagery. Tedlek *et al.* [19] proposed an image processing algorithm to remove cloud from an image. They showed result on a single image with a synthetic, translucent cloud.

### 2.4. Supervised Fusion Methods

These methods either require cloud-free images of the same region or labelled data indicating cloud pixels in images. The method proposed by Mateo *et al.* [12] requires cloud-free images to estimate a background image. A difference image is computed from the cloudy input image and the estimated background is used to predict cloud masks. Even though specific cloud masks are not required, clean images without clouds are required by this work. Method of Li *et al.* [8] proposed a nonnegative matrix factorization based algorithm for cloud removal. This method also requires a cloud-free reference image, which might not be possible. Khan *et al.* [6] present an inpainting method for completion of missing data from satellite images of forests. In this work, the *quality label* for every pixel is assumed to be available. Fusion is done using imagery captured at different times based on the available quality maps and spatial consistency is enforced by solving an optimization problem.

### 2.5. Cloud Invariant Segmentation

Rußwurm and Körner [17] presented an LSTM and GRU-based recurrent system that can use multiple images

simultaneously for vegetation classification from overhead images. Although this work focused on classification, some qualitative results show that the network learns to ignore the cloudy images. In a newer work [16], they provided a quantitative analysis of segmentation results showing that even with an increase in proportion of clouds in multiple images, classification results remain largely the same. This analysis reinforced the claim that the recurrent network had learned to filter out cloudy images. There were still at least four images without clouds, in the worst case.

Our work differs in two ways: first we do not assume the availability of any completely clean images. Secondly, instead of learning to ignore clouds, we present a more general framework that can fuse multiple images to make a single, good image which can then be used for any task. We do so without requiring additional labels for clouds or missing data.

### 3. Weakly-Supervised, Multi-Image Fusion

We present a multi-image fusion method that can take any number of input images to predict a fused image without requiring labels of artifacts in the input images.

#### 3.1. Problem Formulation

For a given geographic region, we assume there exists a set of noisy images,  $I = \{I_1, \dots, I_K\}$ , where  $I_j \in \mathbb{R}^{h \times w \times 3}$ . The goal is to fuse these images so that the combined image  $F = \phi(I)$  is free of artifacts. We assume that semantic segmentation labels (e.g., roads and buildings) of the region are available. Specifically, for each region, we have a segmentation mask  $S \in \mathbb{R}^{h \times w \times C}$ , where  $C$  is the number of classes. Neither examples of clean images nor explicit annotations for artifacts in the input images are available, making this a weakly supervised task.

#### 3.2. Proposed Framework

Our proposed framework is shown in Figure 2. There are three main components: a per-pixel quality prediction network, a fusion module, and a semantic segmentation network. The quality network predicts per-pixel quality for all input images and then a fusion module synthesizes the fused image. The synthesized image is then used as input to the segmentation network. We discuss these components in more detail in subsequent sections.

#### 3.3. Per-Pixel Quality Prediction

Each image  $I_j$  is passed through a quality network which predicts a quality mask,  $Q_j \in \mathbb{R}^{h \times w \times 1}$ , which is a per-pixel mask of logits. Any pixel-wise classification network can be used as the quality network. In this work, we use a variant of the U-Net [14] architecture with the same number of layers but with 1/4 as many feature maps. We found lim-

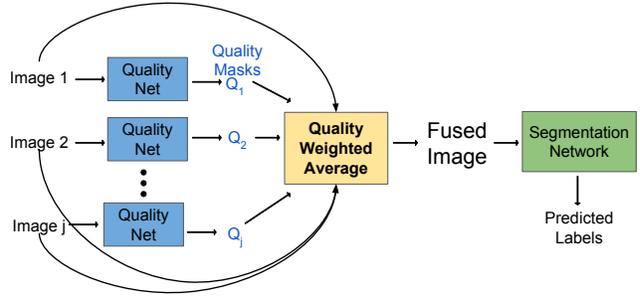


Figure 2: The proposed framework. We train a quality network to predict per-pixel quality of input images and the fused image is synthesized by computing quality-weighted average of input images. The fused image is fed into a segmentation network. We train both networks simultaneously using only the segmentation labels.

ited performance gain if we increase the number of feature maps.

#### 3.4. Image Fusion Module

Once we have pixel-wise quality scores of all input images, we can then compute the relative quality score, at each pixel, by computing the softmax across images:

$$Q_j^*(x, y) = \frac{e^{Q_j(x, y)}}{\sum_{k=1}^K e^{Q_k(x, y)}}. \quad (1)$$

where  $(x, y)$  is the pixel location. The final fused image  $F$  is obtained by averaging all images weighted by the relative quality score:

$$F(x, y) = \sum_{j=1}^K I_j(x, y) \cdot Q_j^*(x, y). \quad (2)$$

Since softmax is a differentiable operation, we can use this operation and train our method in an end-to-end fashion.

#### 3.5. End-to-End Learning

We pass the fused image  $F$  through the segmentation network to get classification labels:  $p = \psi(F)$  so that  $p_i(x, y)$  is the predicted probability of the pixel  $(x, y)$  belonging to class  $i \in \{1, \dots, C\}$ .  $\psi$  can be any arbitrary semantic segmentation network. We use the same variant of the U-Net (with 1/4 number of feature maps) for segmentation. Since we only have segmentation labels as supervision, we use crossentropy loss for end-to-end training:

$$\mathcal{L}(p, S) = -\frac{1}{h \times w} \sum_{x=1}^w \sum_{y=1}^h \sum_{i=1}^C \alpha_i \cdot p_i(x, y) \log(S_i(x, y)) \quad (3)$$

where  $S_i(x, y)$  is the true probability of class  $i$  at  $(x, y)$  and  $\alpha_i$  is a weight factor for every class. These weights can be used to give more importance to some classes or to deal with imbalanced data.

It is important to note that we optimize both the quality network ( $\phi$ ) and the segmentation CNN ( $\psi$ ) with segmentation labels, using the loss function (3). Even though we do not have direct supervision of quality labels, the network learns it implicitly: a good quality prediction leads to better segmentation results. This is similar to many recent works which learn some differentiable operation within the network without direct supervision. For example, spatial transformer networks [4] learn affine transformation, without explicit supervision, that improve the end goal of object recognition.

## 4. Experiments

We now discuss the dataset and implementation details, followed by qualitative and quantitative analyses. We have released our code online <sup>1</sup>.

### 4.1. Dataset

We use data from the City-OSM dataset [5]. In this dataset, aerial imagery is collected from Google Maps and labels are obtained from Open Street Maps (OSM). The labels include road, building and background. We train on images covering  $10.26 \text{ km}^2$  from Berlin. We perform evaluations on  $2.16 \text{ km}^2$  area in Potsdam. The area covered by Potsdam is the same as in the ISPRS Potsdam dataset [15]. The ground sampling distance for both cities is  $9.1 \text{ cm}$ .

Following recent studies on removing clouds from images [18, 19], we make a synthetic dataset. We study the effect of missing data and clouds. Having a synthetic dataset allows the provision of having the *clean* or *ground-truth* images. In this work, we do not train using clean images or even cloud masks: clean images are used only for evaluation of models. A key benefit of a synthetic dataset is the ability to conduct systematic empirical studies, without needing expensive annotations. We examine how the fusion process is effected by variables such as number of input images and amount of artifacts in each image.

We simulate missing data by setting values within a region to zero. We randomly select regions based on the desired area. To synthesize clouds, following existing methods, we use alpha blending. The process of superimposing clouds is shown in Figure 3. We select regions of random sizes at random locations to place clouds. Within each window, we use Perlin noise as a stochastic method of deciding cloud shape, as shown in Figure 3(b). While existing methods use plain white color, we use real cloud images (Figure 3(c)) for alpha blending to get the final cloudy image

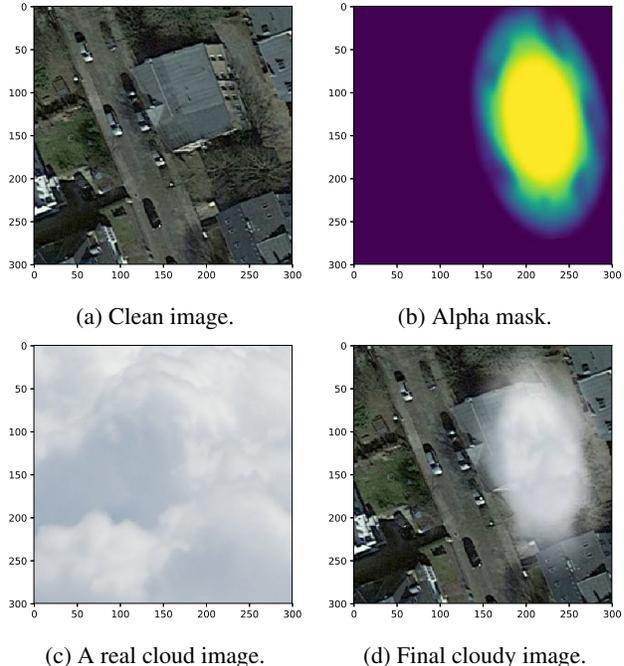


Figure 3: The process of generating cloudy images. We randomly select a region and generate an alpha mask using Perlin noise (b). We use a real cloud image (c) to synthesize the cloudy image (d).

(Figure 3(d)). The real cloud images are split into train and test sets.

### 4.2. Metrics and Evaluation

We evaluate our method in terms of the quality of the fused image and the correctness of the semantic segmentation. For segmentation, we report, mean pixel accuracy, Jaccard index (also known as mean intersection over union (mIoU)), and frequency-weighted intersection over union (fwIoU) [11]. Even though we do not use clean images for training, we show similarity of the fused images with the clean images in terms of mean  $L_1$ -error. For comparison, we use an unweighted average as a fusion strategy. The baseline method includes the same U-Net variant segmentation network as our approach.

To conduct systematic analysis, for every test of a particular number of images and artifact quantity, we train both our model and the baseline method. The baseline method only has the segmentation network while our method trains both the segmentation network as well as the quality network.

To get a rough upper bound on segmentation results, we also train a segmentation network with *clean data* without any artifacts. An ideal fusion system should have segmentation results as good as that of clean data.

<sup>1</sup><http://github.com/UkyVision/weakly-supervised-image-fusion>

	S1: 2 images, 25% area			S2: 6 images, 50% area		
	Acc.	mIoU	fwIoU	Acc.	mIoU	fwIoU
Clean Data	73.25	49.20	59.07	73.25	49.20	59.07
Baseline	69.20	42.78	53.94	61.96	25.15	41.36
Ours	<b>72.91</b>	<b>49.44</b>	<b>58.64</b>	<b>72.15</b>	<b>49.90</b>	<b>57.96</b>

Table 1: Quantitative results on two different scenarios. Clean data show results of a network trained on a single good image without any cloud or occlusion.

### 4.3. Implementation Details

We use the same variant of U-Net with 1/4 number of feature maps for both the quality and segmentation networks. Specifically, in 2D convolutional layers, number of output feature maps are: 32, 64, 128, 128, 64, 32, 16, 16, and 1.

The images in this dataset are large and of different sizes. We split every tile in the original dataset into sixteen images, giving us 3200 images in Berlin. Every image is resized to  $300 \times 300$ . We train our method, and the baseline, on images from Berlin, using 2560 images for training and 640 images for validation. We test our method on 384 images from Potsdam. Since location, orientation, and aspect ratios of artifacts are chosen randomly, we conduct test set evaluation 10 times and present mean values. The dataset has a class imbalance problem: most of the pixels belong to the background class and there are very few road pixels. To deal with this, we use weight factors  $\alpha_i$  of 2, 1, and 0.5 for road, building, and background, respectively.

We train all models for 20 epochs with the Adam [7] optimizer using  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . We set an initial learning rate of  $1 \times 10^{-4}$ , The learning rate is halved after every 5 epochs. We train both networks from scratch with random initialization.

## 5. Results

### 5.1. Quantitative Results

We present segmentation results of our method, baseline, and a network trained on clean data in Table 1. We consider two different scenarios. First, we have two input images and artifacts up to 25% of each image (S1). Secondly, we consider a scenario (S2) in which there are six input images, each having clouds or missing data up to 50% of the total area. In Table 1, we can see that our proposed fusion method outperforms the baseline with a significant margin. We present segmentation results of clean data, as a reference, in the top row.

We are not directly optimizing for the fused images to be closer to the clean images, since we do not assume avail-

	S1: 2 images, 25% area	S2: 6 images, 50% area
	$L_1$ -Error	$L_1$ -Error
Clean Data	0.0	0.0
Baseline	0.2820	0.7608
Ours	<b>0.0177</b>	<b>0.0372</b>

Table 2: Quantitative results on two different scenarios (S1 and S2) on the test set. We present clean data information as a reference, showing that ideal fusion process would lead to mean  $L_1$ -error of zero. We can see that our  $L_1$ -error of our proposed method is 15 to 20 times lower than the baseline.

ability of such data for training. Still, we can see how well our method is doing in terms of synthesizing fused images that are similar to the true, clean images. We present this similarity in terms of mean per-pixel  $L_1$ -error, in Table 2. We can see that our method has significantly lower error than the baseline method.

### 5.2. Varying Number of Input Images

We analyze the effect of number of input images, keeping the area of occlusion and clouds fixed to roughly half of the image. The results are shown in Figure 4. Understandably, the hardest case is with two input images, given that roughly half of each image has artifacts. As we increase number of images, the segmentation results slightly improve (Figure 4(a)). Even though there are some variations in results, overall, the segmentation results are stable at much higher values than the baseline. On the other hand, segmentation results of the baseline method consistently get worse as number of input images increases. The trend is highlighted when we consider  $L_1$ -error between the fused image and the clean image. While our method converges to nearly zero, error in baseline keeps rising, as shown in Figure 4(b).

### 5.3. Increase in Occlusions and Clouds

Next, we evaluate the effect of different proportions of occlusions and clouds in images while fixing the number of images to two. In this study, one image is cloudy and the other is occluded. The performance of our method is compared to the baseline, as shown in Figure 5. We can see that as relative area of artifacts increase in the input images, the results get worse. However, segmentation and reconstruction results of the proposed method significantly outperform the baseline.

### 5.4. Qualitative Results

We now present some qualitative results. First, we show the case of two input images in Figure 6. It can be seen that the relative quality masks ((b) and (d)) can successfully

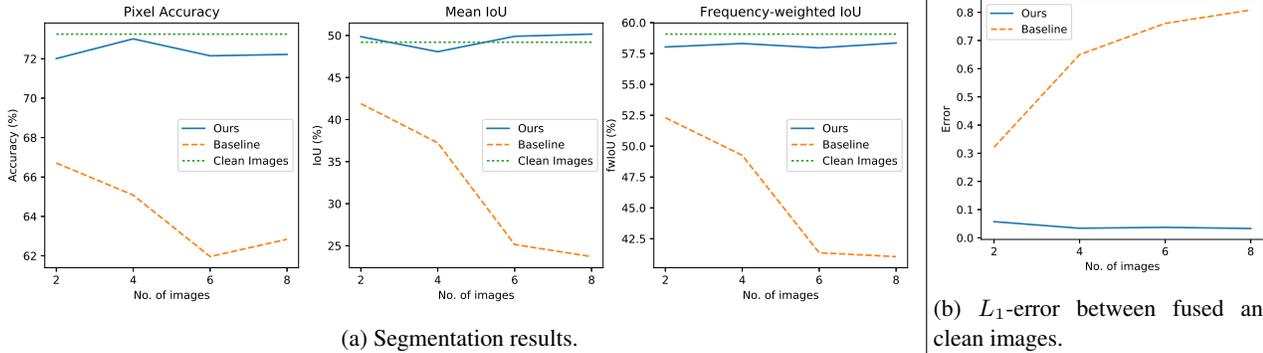


Figure 4: Test set performance of our method vs baseline as input images vary in number. Segmentation results (a) and  $L_1$ -error (b) show the superiority of our method. We can see that our method gets similar results to a network trained with clean images.

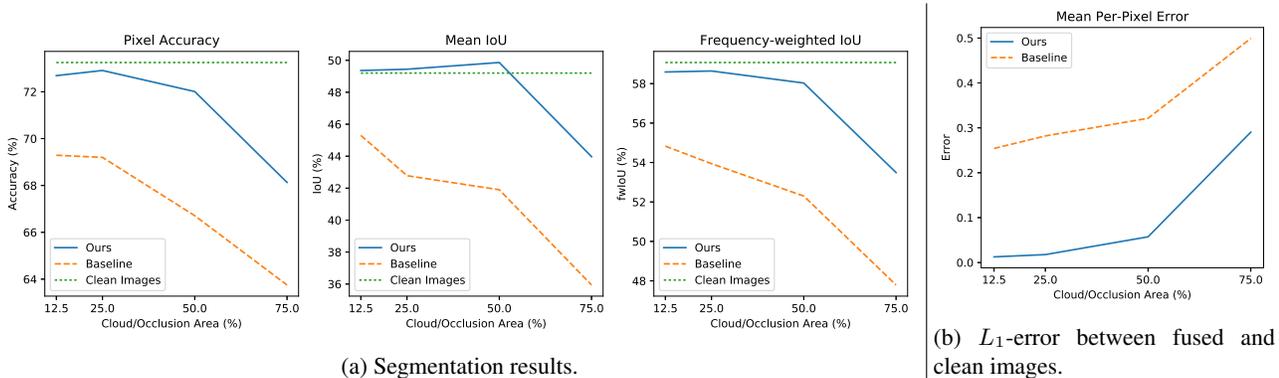


Figure 5: Performance of our method vs baseline, on the test set, as quantities of occlusion and clouds increase. As clouds (and occlusions) increase, the performance of all methods degrades. However, we show that for as large amounts as 50% of image is cloud/occlusion, our method is quite robust.

capture artifacts including clouds and missing data. While artifacts are obvious in the baseline method (c), the fused images (f) very closely match the true, clean images. The last rows shows a challenging case where cloud and missing data overlap. Despite the artifact in the fused image, the result looks much better than the baseline.

We now present the case of four input images. In addition to the fused image, we show the predicted and true segmentation labels.

## 6. Conclusion

We presented a robust fusion method which can learn to combine multiple overhead images of a region. We formulated the problem as a weakly supervised task where only segmentation labels, which are readily available from sources like Open Street Maps, are used for training. We showed that our end-to-end training method gives better segmentation results than the baseline. A major benefit of

our method is that the fused image matches the true representation (in terms of  $L_1$ -error) without requiring direct supervision of artifacts or annotations of clean regions of images. We conducted a systematic analysis to study the effect of varying input images and different proportions of artifacts in each image. For systematic evaluations and proof of concept, we used a synthetic dataset in this work. However, this is a limitation and we plan to work on real data in the future.

## Acknowledgements

We gratefully acknowledge the support of NSF CAREER (IIS-1553116). We are thankful for helpful discussions with Connor Greenwell, Adam Kraft, and Dr. Samson Cheung.

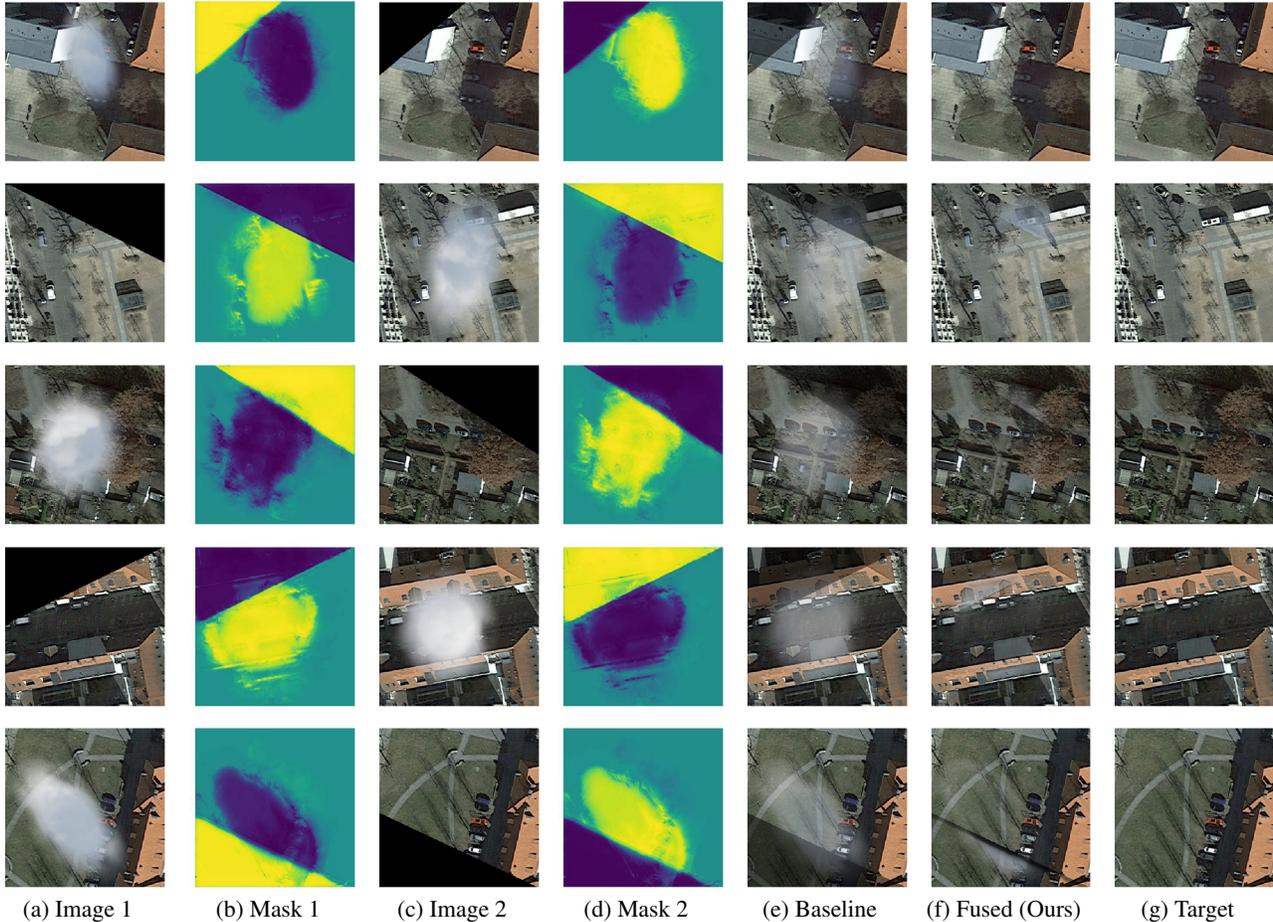


Figure 6: Qualitative results with two input images (a) and (c). We can see that predicted relative quality masks (b) and (d) have learned to locate missing data and clouds. The last row shows a case where it is impossible to synthesize a clean image; performance of our method degrades gracefully and the fused image is still better than the baseline.

## References

- [1] B. Chen, B. Huang, L. Chen, and B. Xu. Spatially and temporally weighted regression: a novel method to produce continuous cloud-free Landsat imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 55(1):27–37, 2017.
- [2] K. Enomoto, K. Sakurada, W. Wang, H. Fukui, M. Matsuoka, R. Nakamura, and N. Kawaguchi. Filmy cloud removal on satellite imagery with multispectral conditional generative adversarial nets. *arXiv preprint arXiv:1710.04835*, 2017.
- [3] S. Foga, P. L. Scaramuzza, S. Guo, Z. Zhu, R. D. Dille, T. Beckmann, G. L. Schmidt, J. L. Dwyer, M. J. Hughes, and B. Laue. Cloud detection algorithm comparison and validation for operational landsat data products. *Remote Sensing of Environment*, 194:379–390, 2017.
- [4] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015.
- [5] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 2017.
- [6] S. H. Khan, X. He, F. Porikli, and M. Bennamoun. Forest change detection in incomplete satellite images with deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9):5407–5423, 2017.
- [7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [8] X. Li, L. Wang, Q. Cheng, P. Wu, W. Gan, and L. Fang. Cloud removal in remote sensing images using nonnegative matrix factorization and error correction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 148:103–113, 2019.
- [9] Z. Li, H. Shen, Q. Cheng, Y. Liu, S. You, and Z. He. Deep learning based cloud detection for remote sensing images by the fusion of multi-scale convolutional features. *arXiv preprint arXiv:1810.05801*, 2018.
- [10] C.-H. Lin, K.-H. Lai, Z.-B. Chen, and J.-Y. Chen. Patch-based information reconstruction of cloud-contaminated multitemporal images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):163–174, 2014.

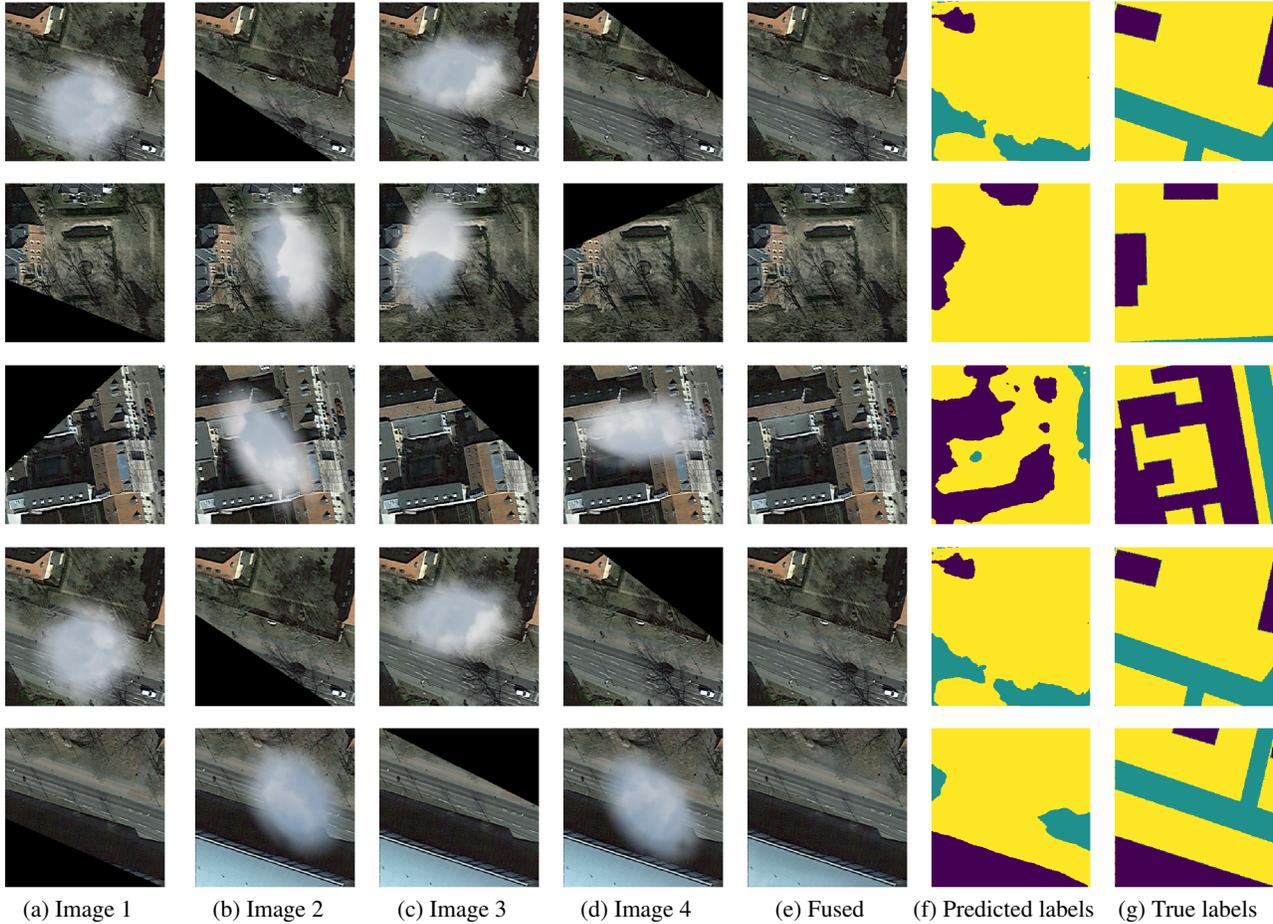


Figure 7: Qualitative results with four input images (a)-(d). Fused image is shown in (e). Predicted and true segmentation labels are shown in (f) and (g), respectively. Last two rows show the imperfect road classification. In all cases, the fused image seems to avoid artifacts in input images.

- [11] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [12] G. Mateo-García, L. Gómez-Chova, J. Amorós-López, J. Muñoz-Marí, and G. Camps-Valls. Multitemporal cloud masking in the Google Earth Engine. *Remote Sensing*, 10(7):1079, 2018.
- [13] G. Roerink, M. Menenti, and W. Verhoef. Reconstructing cloudfree NDVI composites using Fourier analysis of time series. *International Journal of Remote Sensing*, 21(9):1911–1917, 2000.
- [14] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015.
- [15] F. Rottensteiner, G. Sohn, M. Gerke, and J. D. Wegner. ISPRS test project on urban classification and 3D building reconstruction. *Working Group III/4-3D Scene Analysis Commission III-Photogrammetric Computer Vision and Image Analysis*, 2013.
- [16] M. Rußwurm and M. Körner. Convolutional LSTMs for cloud-robust segmentation of remote sensing imagery. *arXiv preprint arXiv:1811.02471*, 2018.
- [17] M. Rußwurm and M. Körner. Multi-temporal land cover classification with sequential recurrent encoders. *ISPRS International Journal of Geo-Information*, 7(4):129, 2018.
- [18] P. Singh and N. Komodakis. Cloud-GAN: Cloud removal for Sentinel-2 imagery using a cyclic consistent generative adversarial networks. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018.
- [19] D. Tedlek, S. Khoomboon, T. Kasetkasem, T. Chanwimaluang, and I. Kumazawa. A cloud-contamination removal algorithm by combining image segmentation and level-set-based approaches for remote sensing images. In *IEEE Conference on Embedded Systems and Intelligent Technology & Conference on Information and Communication Technology for Embedded Systems*, 2018.
- [20] Y. Zhang, B. Guindon, and J. Cihlar. An image transform to characterize and compensate for spatial variations in thin cloud contamination of Landsat images. *Remote Sensing of Environment*, 82(2-3):173–187, 2002.