# Single Image Water Hazard Detection Using FCN with Reflection Attention Units

Xiaofeng Han[1], Chuong Nguyen[2,3,4], Shaodi You[2,3], and Jianfeng Lu[1(✉)]

[1] Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, China
`lujf@njust.edu.cn`
[2] Australian National University, Canberra, ACT 2600, Australia
[3] CSIRO DATA61, Canberra, ACT 2601, Australia
[4] Australian Centre of Excellence for Robotic Vision,
2 George Street, Brisbane 4001, Australia

**Abstract.** Water bodies, such as puddles and flooded areas, on and off road pose significant risks to autonomous cars. Detecting water from moving camera is a challenging task as water surface is highly refractive, and its appearance varies with viewing angle, surrounding scene, weather conditions. In this paper, we present a water puddle detection method based on a Fully Convolutional Network (FCN) with our newly proposed Reflection Attention Units (RAUs). An RAU is a deep network unit designed to embody the physics of reflection on water surface from sky and nearby scene. To verify the performance of our proposed method, we collect 11455 color stereo images with polarizers, and 985 of left images are annotated and divided into 2 datasets: On Road (ONR) dataset and Off Road (OFR) dataset. We show that FCN-8s with RAUs improves significantly precision and recall metrics as compared to FCN-8s, DeepLab V2 and Gaussian Mixture Model (GMM). We also show that focal loss function can improve the performance of FCN-8s network due to the extreme imbalance of water versus ground classification problem.

**Keywords:** Water puddle detection · Road hazard detection
Fully convolutional network · Deep learning · Reflection attention unit

## 1 Introduction

It is well-known that adverse weather conditions affect traffic safety [10,11,20, 30]. Weather-related driving risks are elevated in wet weather not only for human but also for autonomous cars [3,15]. Water and reflection on water surface can cause serious problems in many scenarios. Running into deep water puddle could cause damages to mechanical and electronic parts of the vehicle.

Detection of water puddles on road is, however, not a trivial task because of the wide varieties of appearance and reflection of surrounding environment. Many existing methods rely on special sensors such as dual-polarized cameras [25], near field Radar [2]. However, such devices are not general applicable for normal autonomous cars and are not providing sufficient detection accuracy.

Existing image based detection methods simplify the problem by utilizing multi images along with hand-crafted features, such as average brightness [8], Gaussian fitting of brightness and saturation [16]. However, the water puddle detection are highly ill-posed for hand-crafted features because the real outdoor environments are far too complex to be properly modeled with those hand-crafted features. For example, in Fig. 1, the reflections are coming from sky, clouds and a variety of environment objects.

On the other hand, Deep neural nets can learn features autonomously and have achieved great performance in outdoor navigation nowadays. However, to the best of our knowledge, there is no existing work using deep nets to tackle the water hazard detection. Note that it is not a trivial task because water puddles do not have a well-defined appearance which varies drastically with surrounding environment. Furthermore, there is no existing dataset that is large enough for the training of deep nets.

In this paper, we propose a water detection method based on a Fully Convolutional Network (FCN) with reflection attention units (RAUs). The RAUs are designed to allow the network to capture reflection correspondences between different parts of the images. Because the reflection correspondences are mostly vertical, feature maps in multi-scales are divided into several patches along vertical directions. Then average of each patch is calculated. All the pixels are compared with the averages in the same column to determine whether it is a reflection of a certain patch. As shown in Fig. 1(c), since the $X_8$ is the reflection of $X_6$, the subtraction results between pixels in $X_8$ and the average of $X_6$ should be lower than that of other pairs. Figure 1 also shows the water hazard detection results of our method (d) and FCN-8s (e). Because the reflections on the water surface are detected by (c), our method clearly outperforms FCN-8s. In addition we propose to replace cross entropy loss function by focal loss [12] to deal with the data imbalance problem, as the size of the water puddle various tremendously between images.

Furthermore, in order to verify the performance of our proposed method and encouraging new research, we propose the 'Puddle-1000' Dataset. We collect 11455 color stereo images with polarizers, and 985 of left images are annotated and divided into 2 datasets: On Road (ONR) sub-dataset and Off Road (OFR) sub-dataset.

As far as we know, this is the first work to exploit deep neural networks on water hazard detection. And our main contributions are as follows:

– We propose a reflection attention unit (RAU), and insert it after every last convolutional layer of 5 group layers in the FCN-8s network [13]. These units are designed to pick up reflection correspondences relationships between different vertical parts of images.
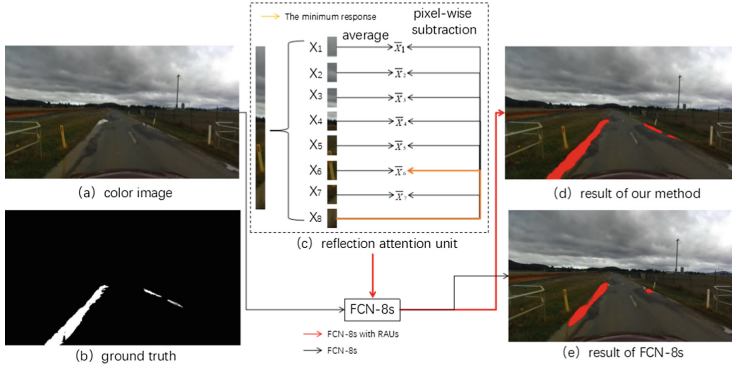
**Fig. 1.** Single image water hazard detection using a deep net with and without RAUs. (c) illustrates the proposed reflection attention unit (RAU) to automatically learn the reflection relationship within a single image. (Color figure online)

- We take the advantage of focal loss [12] to deal with this imbalanced water detection problem where water puddles account for a small fraction of total number of pixels.
- To the best of knowledge, we propose the first single image deep neural net based method on water hazard detection in real driving scenes. And the proposed method achieves the state-of-the-art performance.
- We have pixel-wise annotated 985 images mostly containing water puddles. These include 357 on road images and 628 off road images. This newly annotated dataset and source codes of deep networks are available to public.

## 2 Related Work

**Traversable Area Detection and Semantic Segmentation Deep Neural Networks**
Water hazard detection is often considered as complementary to the traversable area detection, or the road detection. Traditional methods are based on hand-crafted features and color priors. For example, Lu et al. [14] used Gaussian Mixture Model (GMM) to estimate the priors.

Recently, CNN based semantic segmentation methods have demonstrated a superior performance. Long *et al.* [13] first proposed a fully convolutional network (FCN). And after that, by taking the benefit from ResNet [7], Chen *et al.* [4] proposed the Deeplab and futher improved the performance. Zhao *et al.* [31] proposed a new network structure, called pyramid pooling module, to exploit global context information. Recently, Han *et al.* [6] proposed a semi-supervised learning semantic segmentation method based on generative adversarial network (GAN). However, the hyper-parameters of GAN were selected empirically which are not robust to various water detection scenes.

**Puddle Detection**

*Active Imaging Based Methods:* To determine different road surface conditions such as dry, wet, snowy and frozen, Fukamizu *et al.* [5] relied on the reflectance of a projected visible and infrared light source. Viikari *et al.* [25] proposed to use the backscattering of dual polarized RADARs, while Bertozzi *et al.* [2] used backscattering of short infrared light source. These techniques however have limited working range which is up to 2 m. For larger distance, images extracted from cameras are required.

*Single Image Method:* Zhao *et al.* [32] exploits the water region detection using color constancy and texture. However they assume a close-up image of water puddles with uniform reflection of the sky. Such assumptions simply do not hold in real driving conditions.

*Stereo Images/Video Based Methods:* Texture cues of water surface have been used including smoothness descriptor [29], and local binary pattern [17,18]. Color cue in HSV space have also been used by Yan [29]. Rankin and Matthies [21] further showed that color of water puddle varies with viewing distance and angle. Temporal fluctuation (or dynamic texture) of water surface was also exploited by Santana *et al.* [23] (optical flow), and Mettes *et al.* [17,18] (temporal deviation and L1-norm FFT) to successfully detect running water bodies or still water under windy condition from a fixed camera position. Stereo depth was also exploited by Matthies *et al.* [16], Yan [29], and Kim *et al.* [9]. As light reflected from water surfaces is polarized [26], this provides a strong cue to detect water puddles as used in several works [9,19,24,28,29]. While Xie *et al.* [28] used 3 cameras, others used stereo cameras attached to horizontal and vertical polarizers. Nguyen *et al.* [19] further showed that sky polarization strongly effect the appearance of water.

Other imaging wavelengths including infrared and thermal imaging are also used by [2,16,22] to allow water detection at any time without active light sources. Rankin *et al.* [22] also showed that the relative intensity of water versus ground changes distinctively between night and day and this provide a strong cue of water.

**Classification Techniques**

From obtained cues of water, different classification techniques have been utilized including hard coded and adaptive thresholding [9,21,28,29], K-means [23], decision forest [17], support vector machine [9], and Gaussian mixture model [19].

To exploit temporal constraint, state propagation techniques have been used including segmentation guided label [23] and Markov random field [17].

## 3    Problem Formation and Physical Insights

In this section, we formulate the problem and explicitly explain why water puddle detection in a single image is a challenging problem. Also, we introduce concept on reflection attention.
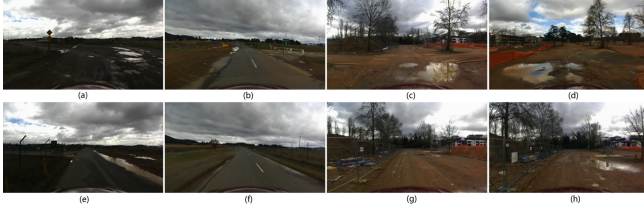
**Fig. 2.** Examples from the proposed 'Puddle-1000' dataset. Water on ground could have different reflections, colors, brightness, and shapes. Water surface can be still or moving. (Color figure online)

**Appearance of Water Puddle: Reflection.** Detecting water hazard from distance is challenging due to the very nature of reflection on water surface. Examples of water are given in Fig. 2. In (a) the shape and boundary of puddles are very irregular. In (b) even though there are only reflections of sky, the colors of water surfaces change with the distances. Both in (a) and (b), reflections of puddles far away are very bright. In (c) the reflections are mainly about clouds and trees, while in (d) they consist of blue sky, red fences, clouds and trees, which makes the puddles look very different. The colors of the puddles in (a) and (b) are different from those in (c) and (d), as the latter ones have more soil sediments. In (e) the right puddle has waves due to the wind. The left puddle in (f) looks very similar with road areas. (g) and (h) contain the same puddle but captured at different distances. The puddle reflects of different parts of trees, therefore the textures on puddle surface are quite different.

**Appearance of Water Puddle: Inter Reflection/Refraction and Scattering.** The process of light reflecting and scattering from a water puddle is illustrated in Fig. 3. Modeling the appearance of a puddle with a provided environmental luminance is, yet, ill-posed. Light source $S_1$ from the sky or nearby objects hit water surface at $O_1$. It partially reflects back into the air and partially refracts into water column as shown in the left of the figure. What we see from the water puddle is a summation of (a) reflection of the light source as at $O_1$, and (b) the fraction from inside the water as at $O_2$ and $O_3$. Reflection as shown in the top right in fact is the combination of specular reflection (i.e. clear image) $R_{reflect}$ and scatter reflection (i.e. blurry image) $R_{scatter}$. Similarly, the refraction as shown in the bottom right is the combination of light coming straight from the ground bottom $R_{bottom}$ as at $S_3$ and light from sediment particles $R_{particles}$ as at $S_2$. This process is expressed as the following summation:

$$R_{total} = R_{reflect} + R_{scatter} + R_{bottom} + R_{particles} \tag{1}$$

An important property of reflection is that the light source and its reflected image lie on a straight line perpendicular to the water surface (or the ground) and that they have same height from the ground as shown in Fig. 4. Perspective view and imperfect camera lens introduce some distortions to captured images. The line connecting the source and its reflection may not be exactly vertical in
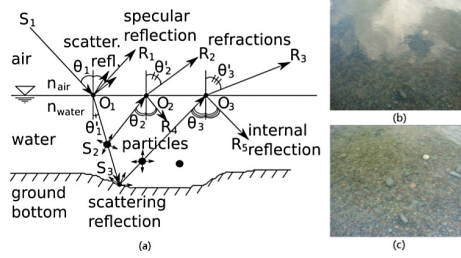
**Fig. 3.** (a) shows the process of water reflection ($S_1O_1R_1$), refraction ($S_2O_2R_2$ and $S_3O_3R_3$) and scattering ($O_1S_2O_2$ and $O_1S_3O_3$) at a water puddle. (b) and (c) show light components from the same water puddle. (b) shows reflection light on water surface, magnified by passing through horizontal polariser. (c) shows fraction light from inside water puddle, magnified passing through vertical polariser (wikipedia.org).
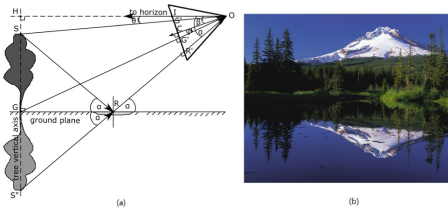


**Fig. 4.** (a) shows image formation of an object and its reflection. The light source and its image is on a line SS" perpendicular at ground G where GS = GS". However the pixel distances on camera image are different, G'S' ≠ G'R'. (b) shows reflection on water with vertical correspondences between tree tops and their reflections (wikipedia.org).

the image, and the distances of the source and its reflection to the ground are not exactly the same. As the distance from the camera increases, the different in the height of the object and its reflection reduces. We aim to design a deep network that captures this reflection effect and tolerates the distortion and camera rotation.

**Mining Visual Priors Through Deep Learning.** With recent rapid advancements of Convolutional Neural Network (CNN) to effectively solve various traditional computer vision problems, we aim to apply this powerful tool to the problem of water hazard detection. CNN in general and Fully Convolutional Network (FCN) in particular recognise objects with distinct structures and patterns. Therefore these networks do not work well with water reflection which varies drastically depending on what is reflected. As a result, we want to extend the networks recognise the physics of the reflection phenomenon. The main characteristics of reflection on water surface is that the reflection is a inverted and disturbed transform of the sky or nearby objects above the water surface. Specifically, we propose in this paper a new network module called Reflection Attention Unit (RAU) that matches image pattern in the vertical direction.

# 4  Fully Convolutional Network with Reflection Attention Units

Water hazard is hard to recognize for existing semantic segmentation methods because of the reflections on the water surface. Therefore, we want to exploit more information about local and global reflection contexts especially in the vertical direction. The proposed Reflection Attention Units (RAUs) are then used to learn the reflection relationships.

## 4.1  Reflection Attention Unit

We propose RAU based on a strong cue in a single image that water puddles usually contain vertical reflections. As illustrated by Fig. 4, a reflection and its source lies along a line nearly vertical in the image. Therefore to detect water puddles, we can search for reflections by matching image regions along pixel columns of an image. Furthermore, to tolerate perspective distortion, small camera rotation (angle with line of horizon) and blurry reflection, multiple resolutions or scales are used in the vertical matching.

The architecture of the proposed RAU is illustrated in Fig. 5. Given an input feature map $I$ of size $[h, w, c]$, horizontal average pooling is applied to $I$ reduce to size $[h, w/2, c]$. Then vertical average pooling is applied to reduce this to $X$ of size $[n, w/2, c]$. In Fig. 5, $n$ is set to be 8 for illustration purpose. After that, each row $X_i$ sized $[1, w/2]$ of $X$ is tiled or self replicated to size $[n, w/2]$. Those obtained from all rows are concatenated along the feature axis into a new feature map of size of $[n, w/2, c*n]$. Then, this feature map are up-sampled to size $[h, w, c*n]$ and denoted as $X'$. We then concatenate n times of $I$ along the feature axis to get $I'$ with size $[h, w, c*n]$. $I'$ is subtracted from $X'$ to produce $D$ with size $[h, w, c*n]$ which encode the reflection relationships. The subtracted feature map is concatenated with $I$ again, fed into a convolutional layer and activated by ReLU function to generate the final output of the same size as $I$.

Figure 6 illustrate how the RAU computes the similarities between one pixel and the averages of other different parts along vertical direction in the neighboring 2 columns in a certain scale. Take the first channel of $X'$ as an example. This single-channel image has 16 tiled rows and pixels along each column are the same. A single row represents the blurry version of the top rows of the first channel of $I$. Because the top rows of $I$ contain mostly clouds, therefore in the difference $D$, the clouds and reflection of clouds on the water surface has lower intensity than ground, fences and trees. Furthermore, as the reflection lines are not strictly vertical due to distortion and image rotation, the two average poolings allow for such misalignments from these. In addition, such misalignments are also taken account for when several RAUs are applied to feature maps of different scales as outputs of successive convolutional layer groups.
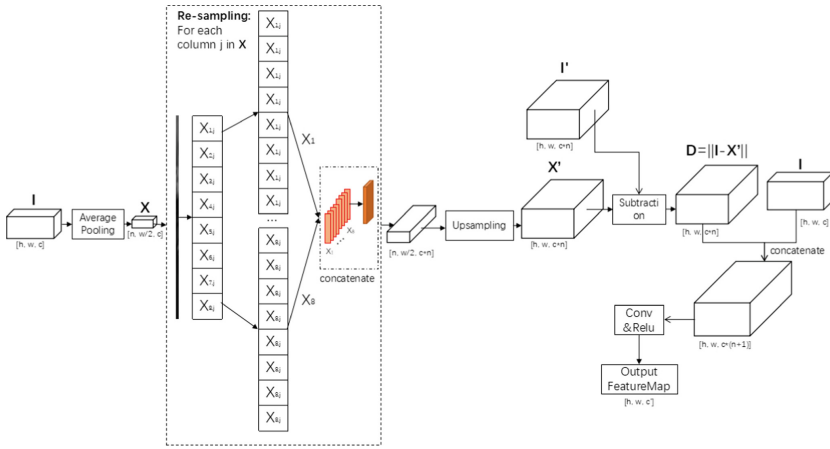
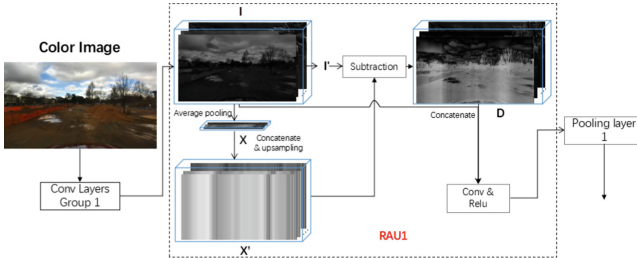**Fig. 5.** Illustration of a Reflection Attention Unit (RAU).



**Fig. 6.** Working principle of a Reflection Attention Unit. After the convolutional layer group 1, the input color image is transformed to a feature map $I$, and average pooling is applied to $I$ to get the downsampled feature map $X$. Then, self replication, concatenation and upsampling of $I$ produce feature map $X'$. $I$ are concatenated to get $I'$. $I'$ is subtracted from $X'$ to produce a difference map $D$. Finally, $D$ and $I$ are concatenated and fed into a convolutional layer and a ReLU to output a new feature map of the same size as $I$. This is fed to pooling layer 1 of a normal convolutional layer group. (Color figure online)

## 4.2   Network Architecture

Figure 7 shows the network architectures of standard FCN-8s [27] (a) and our method (b). To study the usefulness of different RAUs in a FCN-8s network, we increasingly insert 5 RAUs after each group of convolutional layers to enable the reflection awareness at different scales.

## 4.3   Focal Loss

The area of water puddles is found much smaller than that of the ground. Particularly, in ONR dataset and OFR dataset, introduced in the next section, the
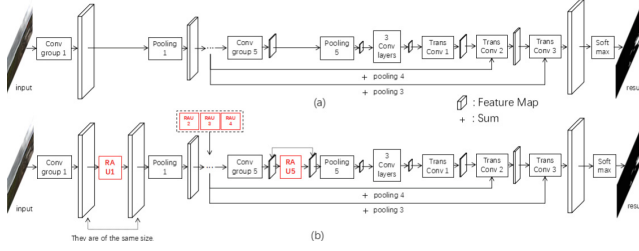
**Fig. 7.** Architectures of FCN-8s and our proposed FCN-8s with RAUs. For compactness, we only show 2 out of 5 groups of convolutional layers and their corresponding RAUs.

ratios between the pixel number of water and non-water are approximately 1:61 and 1:89. These lead to skewed training when using a common loss function, such as cross entropy loss.

To deal with unbalanced classification, we propose to use focal loss [12] with our network. Given such a binary classification problem, we define $y = \{0, 1\}$ as the ground truth classes and $p$ as the probability that one sample belongs to class $y = 1$. Then standard cross entropy loss is defined as follows:

$$CE(p, y) = \begin{cases} -log(p) & if\ y = 1 \\ -log(1 - p) & otherwise \end{cases} \quad (2)$$

If we define $p_t$ as follows:

$$p_t = \begin{cases} p & if\ y = 1 \\ 1 - p & otherwise \end{cases} \quad (3)$$

Then Eq. 2 can be rewritten as:

$$CE(p, y) = CE(p_t) = -log(p_t) \quad (4)$$

The focal loss is defined as follows to down-weight easy examples and focus on hard examples:

$$FL(p_t) = -(1 - p_t)^{\gamma} log(p_t) \quad (5)$$

where $-(1 - p_t)^{\gamma}$ is the modulating factor and $\gamma \geq 0$ is a tunable focusing parameter. $\gamma$ is chosen to be 2 in this paper.

## 5 Puddle-1000 Dataset

To enable deep learning based methods and systematical tests, in this paper we present a new and practical water puddle detection dataset. Note that a previous dataset of water puddles was published by [19] recorded at two different locations for on road and off road conditions near Canberra city, Australia. This dataset only contains 126 and 157 annotated left frames for on road sequence (ONR) and off road sequence (OFR). For this paper, the annotated frame are too limited for training deep networks.
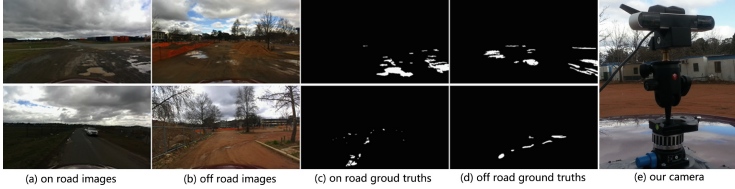
(a) on road images    (b) off road images    (c) on road groud truths    (d) off road ground truths    (e) our camera

**Fig. 8.** Examples in the proposed dataset and the ZED camera. (Color figure online)

**Proposed Dataset.** In this paper, we extend the existing dataset with 5 times more pixel-wise labeled images. The labeled data are all from the above dataset captured by the ZED stereo camera, and since we aim for a single image solution, only left images are annotated and used to validate the performance of different networks. Specifically, ONR now has 357 annotated frames, while OFR now has 628 annotated frames. Figure 8(a) and (b) show color images from ONR and OFR datasets, and (c) and (d) show examples of the pixel-wise annotation masks with two classes. In the ONR dataset, the waters are very muddy and the reflections are mainly from sky, clouds, pillars and fences, while in the OFR dataset the water surfaces are the combined reflections of blue sky, clouds, trees, telegraphs, buildings and some fences. The appearances of ground and other obstacles in those two datasets are also much different. In ONR, there are asphalt roads with moving cars and containers. In OFR the grounds are just wet dirt roads, however it has more different kinds of obstacles, such as fences, buildings, many kinds of trees, mounds and building materials. Therefore there are significant differences in water reflections. These new annotated frames are also released to the public.

## 6    Experiments

We systematically evaluate the proposed network using the proposed dataset. We compare with existing single image based methods and also compare with a various of existing network structures. We also provide detailed analysis on training time, robustness to over-fitting in the supplementary material (because of the length limit.)

### 6.1    Implementation Details

Our network is implemented using Tensorflow [1] framework and is trained on an NVIDIA TITAN XP GPU with 12 GB of memory. In experiments, the resolution of images and ground truths are downsampled to 360px × 640px. The batch size is 1 during training. Learning rate is set to $10^{-6}$ at first, and decreases by a factor of 0.2 every 5K iterations after 20K iterations. The number of training iteration of both FCN-8s and FCN-8S-focal-loss is 100K, and that of both Deeplab-V2 and our proposed network is 60K.

**Table 1.** Details of RAUs

| Name | N | Convolutional kernel | | | Feature map size |
|------|---|----------------------|---|---|------------------|
|      |   | Input channels | Output channels | kernel size | |
| RAU1 | 16 | $64 * (16 + 1)$ | 64 | $[3, 3]$ | $360 \times 640$ |
| RAU2 | 16 | $128 * (16 + 1)$ | 128 | $[3, 3]$ | $180 \times 320$ |
| RAU3 | 16 | $256 * (16 + 1)$ | 256 | $[3, 3]$ | $90 \times 160$ |
| RAU4 | 16 | $512 * (16 + 1)$ | 512 | $[3, 3]$ | $45 \times 80$ |
| RAU5 | 8 | $512 * (8 + 1)$ | 512 | $[3, 3]$ | $23 \times 40$ |

**Table 2.** Performances of the networks with different number of RAUs.

| Dataset | Method | F-measure | Precision | Recall | Accuracy |
|---------|--------|-----------|-----------|--------|----------|
| ONR | FCN-8s-FL-5Conv | 55.76% | 55.68% | 55.85% | 99.06% |
|     | FCN-8s-FL-1RAU | 59.63% | 61.64% | 57.75% | 99.17% |
|     | FCN-8s-FL-3RAU | 61.97% | 63.43% | 60.57% | 99.20% |
|     | FCN-8s-FL-5RAU | **70.11%** | **67.78%** | **72.61%** | **99.35%** |
| OFR | FCN-8s-FL-5Conv | 78.56% | 89.33% | 70.11% | 99.32% |
|     | FCN-8s-FL-1RAU | 71.67% | 87.45% | 60.71% | 99.14% |
|     | FCN-8s-FL-3RAU | 79.09% | **91.36%** | 69.72% | 99.34% |
|     | FCN-8s-FL-5RAU | **81.67%** | 87.21% | **76.79%** | **99.38%** |
| BOTH | FCN-8s-FL-5Conv | 63.44% | 62.52% | 64.44% | 99.21% |
|      | FCN-8s-FL-1RAU | 67.64% | 74.66% | 61.82% | 99.14% |
|      | FCN-8s-FL-3RAU | 67.63% | 75.27% | 64.40% | 99.15% |
|      | FCN-8s-FL-5RAU | **76.91%** | **78.03%** | **75.81%** | **99.34%** |

The ONR and OFR datasets are randomly divided into training and testing categories in the following experiments. For ONR dataset, we use 272 images to train the networks, and 85 images to verify the performances. As for OFR dataset, 530 images are used for training and 98 images for testing. Furthermore, we also carry out the experiments on both datasets combined together. The metrics used for evaluation are F-measure, precision, recall and accuracy. In all experiments, we do not use data augmentation.

The details of 5 RAUs are shown in Table 1.

### 6.2 Validation of the Reflection Attention Units

We train four different networks on the dataset. Three of them are FCN-8s with RAUs, the difference is that, the number of RAUs in these networks are 1, 3 and 5 respectively. They are named as FCN-8s-FL-1RAU, FCN-8s-FL-3RAU and FCN-8s-FL-5RAU. In FCN-8s-FL-1RAU the single RAU is placed before the first pooling layer. In FCN-8s-FL-3RAU the RAUs are added before the first, the third

**Table 3.** Performances of FCN-8s with and without focal loss.

| Dataset | Method | F-measure | Precision | Recall | Accuracy |
|---|---|---|---|---|---|
| ONR | FCN-8s | 56.99% | **59.01%** | 55.11% | **99.12%** |
| | FCN-8s-FL | **57.85%** | 50.49% | **67.71%** | 98.96% |
| OFR | FCN-8s | 64.33% | 78.18% | 54.64% | 98.92% |
| | FCN-8s-FL | **74.05%** | **84.88%** | **65.66%** | **99.18%** |
| BOTH | FCN-8s | 65.21% | 69.81% | 61.18% | 99.05% |
| | FCN-8s-FL | **70.62%** | **74.38%** | **67.22%** | **99.19%** |

**Table 4.** Performances of different methods on cross dataset validations.

| Testing dataset | Training dataset | Method | F-measure | Precision | Recall | Accuracy |
|---|---|---|---|---|---|---|
| ONR | OFR | FCN-8s-FL | 9.16% | 15.80% | 6.45% | 98.65% |
| | | FCN-8s-FL-RAU | **31.43%** | **50.99%** | **22.72%** | **98.95%** |
| OFR | ONR | FCN-8s-FL | 22.98% | 27.20% | 19.89% | 97.62% |
| | | FCN-8s-FL-RAU | **36.60%** | **60.71%** | **26.20%** | **98.38%** |

and the fifth pooling layers. And FCN-8s-FL-5RAU is the proposed network. In the last network we do not use RAUs and only add 5 more convolutional layers. In all the networks we use the focal loss. Table 2 shows that, the performances are improved with increasing using of RAUs.

### 6.3    Validation of the Focal Loss

We train the FCN-8s with the cross entropy loss function and the focal loss, respectively. The results are shown in Table 3. From this table, we can see that using focal loss can get obviously better performances on OFR and BOTH datasets. The reason why there is no significant improvement on ONR dataset is because the data imbalance of ONR dataset is much smaller than that of OFR dataset. Even so, the FCN-8s-FL still gets better F-measure and Recall.

### 6.4    Cross Dataset Validation

To further validate the robustness of our method, we train two networks on ONR and OFR dataset, and verify them on the other datasets. The results (Table 4) of our method are better than the FCN-8s-FL in all the experiments. This indicates our method has much better generalization performance on different datasets.

### 6.5    Comparison with Existing Methods

For further comparison, we implement and re-train other image segmentation networks including FCN-8s, FCN-8s-FL, FCN-8s-5Conv and DeepLab (version 2) [4]. We also compared our method with the non deep learning method,

**Table 5.** Performance comparison between our proposed network and others.

| Dataset | Method | F-measure | Precision | Recall | Accuracy | Time |
|---------|--------|-----------|-----------|--------|----------|------|
| ONR | FCN-8s-FL-RAU (ours) | **70.11**% | **67.78**% | 72.61% | **99.35**% | 0.32 s |
| | FCN-8s-FL | 57.85% | 50.49% | 67.71% | 98.96% | 0.06 s |
| | FCN-8s [27] | 56.99% | 59.01% | 55.11% | 99.12% | 0.06 s |
| | DeepLab [4] | 21.97% | 37.18% | 15.60% | 98.83% | 0.27 s |
| | FCN-8s-FL-5Conv | 55.76% | 55.68% | 55.85% | 99.06% | 0.07 s |
| | GMM & polarisation [19] | 31.0% | 18.7% | **90.2**% | 96.5% | NA |
| OFR | FCN-8s-FL-RAU (ours) | **81.67**% | 87.21% | 76.79% | **99.38**% | 0.32 s |
| | FCN-8s-FL | 74.05% | 84.88% | 65.66% | 99.18% | 0.06 s |
| | FCN-8s [27] | 64.33% | 78.18% | 54.64% | 98.92% | 0.06 s |
| | DeepLab [4] | 45.05% | 71.31% | 32.92% | 98.56% | 0.27 s |
| | FCN-8s-FL-5Conv | 78.56% | **89.33**% | 70.11% | 99.32% | 0.07 s |
| | GMM & polarization [19] | 28.1% | 16.8% | **85.4**% | 95.2% | NA |
| BOTH | FCN-8s-FL-RAU (ours) | **76.91**% | **78.03**% | **75.81**% | **99.34**% | 0.32 s |
| | FCN-8s-FL | 70.62% | 74.38% | 67.22% | 99.19% | 0.06 s |
| | FCN-8s [27] | 65.21% | 69.81% | 61.18% | 99.05% | 0.06 s |
| | DeepLab [4] | 30.36% | 53.52% | 21.19% | 98.59% | 0.27 s |
| | FCN-8s-FL-5Conv | 63.44% | 62.52% | 64.44% | 99.21% | 0.07 s |
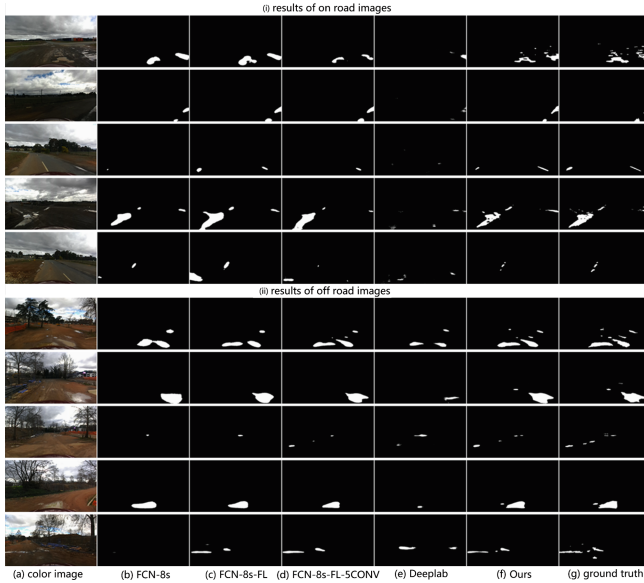


**Fig. 9.** Water hazards detection results trained on both datasets. (Color figure online)
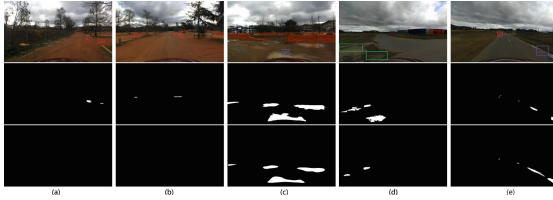
**Fig. 10.** The challenging cases for water hazard detection by our method. The first row are the color images, the second row are the ground truths and the last row are our results. (Color figure online)

such as GMM&polarisation [19]. For DeepLab, we fine-tune it based on a pre-trained Resnet101 model provided by Deeplab, and we do not apply CRF after the inference.[1]

Table 5 shows the performances of different methods and the average inference time for one frame. Figure 9 demonstrates the water hazard detection results of them. We can see that our RAUs help improve the performances a lot. The precision and recall have great improvements, showing that our RAUs can help the networks to reduce the false-positives and false-negatives. Figure 9 also demonstrates that improvements too.[2]

## 7     Discussion and Conclusion

**Challenging Cases.** Even with the help with RAUs, the water hazard detection is still very challenging in various cases. Figure 10 shows some examples. As shown in red rectangles, some puddle areas are too small to be recognized, because they only contain a few pixels. Besides, the wet areas are very similar with puddles, as we present in blue rectangles in (c) and (e). Lastly, in (d) the green rectangles show that some of the water surfaces almost look the same with the road. In all these cases, the water surfaces contain very little reflection information, so that our RAUs can not improve the performances.

**Conclusion.** We propose a robust single image water hazard detection based on fully convolutional networks with reflection attention units (RAUs) and focal loss. We also collect on road and off road color images with water hazards, and pixel-wisely annotate 985 images of them to build a dataset and verify the performances. We apply RAUs on multi-scale feature maps. In this novelly proposed RAUs, we calculate the distances between one pixel and the averages

---

[1] We respectfully mention that we don't have access to source codes and datasets from some other previous publications. And methods not working with single images are not compared too.

[2] The accuracy does not increase significantly because the number of water pixels is much smaller than that of non-water pixels, however, when we calculate the accuracy, we count the detection accuracies both of water and non-water pixels.

of different patches in each 2 columns along vertical direction. The focal loss is also used to deal with the serious data imbalance. Experiments of several deep neural networks and one traditional method on these datasets are carried out, and the results show the great effectiveness of our proposed method.

# References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous distributed systems (2016)
2. Bertozzi, M., Fedriga, R.I., D'Ambrosio, C.: Adverse driving conditions alert: investigations on the SWIR bandwidth for road status monitoring. In: Petrosino, A. (ed.) ICIAP 2013. LNCS, vol. 8156, pp. 592–601. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41181-6_60
3. Byrne, K.: Self-driving cars: will they be safe during bad weather? (2016). https://www.accuweather.com/en/weather-news/self-driving-cars-will-they-be-safe-during-bad-weathr/60524998
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS. arXiv:1606.00915 (2016)
5. Fukamizu, H., Nakano, M., Iba, K., Yamasaki, T., Sano, K.: Road surface condition detection system, 1 September 1987. US Patent 4,690,553
6. Han, X., Lu, J., Zhao, C., You, S., Li, H.: Semi-supervised and weakly-supervised road detection based on generative adversarial networks. IEEE Signal Process. Lett. (in press)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition, pp. 770–778 (2015)
8. Iqbal, M., Morel, O., Meriaudeau, F., Raya, J.M., Gunadarma, U., Fakultas, I.: A survey on outdoor water hazard detection. Institus Teknologi Sepuluh November **33**, 2085–1944 (2009)
9. Kim, J., Baek, J., Choi, H., Kim, E.: Wet area and puddle detection for advanced driver assistance systems (ADAS) using a stereo camera. Int. J. Control., Autom. Syst. **14**(1), 263–271 (2016)
10. Koetse, M.J., Rietveld, P.: The impact of climate change and weather on transport: an overview of empirical findings. Transp. Res. Part D: Transp. Environ. **14**(3), 205–221 (2009)

11. Li, Y., You, S., Brown, M.S., Tan, R.T.: Haze visibility enhancement: a survey and quantitative benchmarking. ComComput. Vis. Image Underst. (2016)
12. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. arXiv preprint arXiv:1708.02002 (2017)
13. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
14. Lu, K., Li, J., An, X., He, H.: A hierarchical approach for road detection. In: IEEE International Conference on Robotics and Automation, pp. 517–522 (2014)
15. Marshall, A.: To let self-driving cars go anywhere, train them everywhere (2017). https://www.wired.com/story/waymo-self-driving-michigan-testing/
16. Matthies, L.H., Bellutta, P., McHenry, M.: Detecting water hazards for autonomous off-road navigation. In: Unmanned Ground Vehicle Technology V, vol. 5083, pp. 231–243. International Society for Optics and Photonics (2003)
17. Mettes, P., Tan, R.T., Veltkamp, R.: On the segmentation and classification of water in videos. In: 2014 International Conference on Computer Vision Theory and Applications (VISAPP), vol. 1, pp. 283–292. IEEE (2014)
18. Mettes, P., Tan, R.T., Veltkamp, R.C.: Water detection through spatio-temporal invariant descriptors. Comput. Vis. Image Underst. **154**, 182–191 (2017)
19. Nguyen, C.V., Milford, M., Mahony, R.: 3D tracking of water hazards with polarized stereo cameras. In: International Conference on Robotics and Automation (ICRA), pp. 5251–5257. IEEE (2017)
20. Office of Operation: How do weather events impact roads? (2017). https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm
21. Rankin, A., Matthies, L.: Daytime water detection based on color variation. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 215–221. IEEE (2010)
22. Rankin, A.L., Matthies, L.H., Bellutta, P.: Daytime water detection based on sky reflections. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 5329–5336. IEEE (2011)
23. Santana, P., Mendonça, R., Barata, J.: Water detection with segmentation guided dynamic texture recognition. In: 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1836–1841. IEEE (2012)
24. Sarwal, A., Nett, J., Simon, D.: Detection of small water-bodies. Technical report, DTIC Document (2004)
25. Viikari, V.V., Varpula, T., Kantanen, M.: Road-condition recognition using 24-GHz automotive radar. IEEE Trans. Intell. Transp. Syst. **10**(4), 639–648 (2009)
26. Wehner, R.: Polarization vision – a uniform sensory capacity? J. Exp. Biol. **204**(14), 2589–2596 (2001)
27. Wu, X.: Fully convolutional networks for semantic segmentation. In: Computer Science (2015)
28. Xie, B., Pan, H., Xiang, Z., Liu, J.: Polarization-based water hazards detection for autonomous off-road navigation. In: 2007 International Conference on Mechatronics and Automation, pp. 1666–1670. IEEE (2007)
29. Yan, S.H.: Water body detection using two camera polarized stereo vision. Int. J. Res. Comput. Eng. Electron. **3**(3) (2014)

30. You, S., Tan, R., Kawakami, R., Mukaigawa, Y., Ikeuchi, K.: Adherent raindrop modeling, detection and removal in video. IEEE Trans. Pattern Anal. Mach. Intell. **38**(9), 1721–1733 (2016)
31. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6230–6239 (2017)
32. Zhao, Y., Deng, Y., Pan, C., Guo, L.: Research of water hazard detection based on color and texture features. Sens. Transducers **157**(10), 428 (2013)