
Attention OCR


- Wojna, Zbigniew, et al. "Attention-based extraction of structured information from street view imagery." 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). Vol. 1. IEEE, 2017.

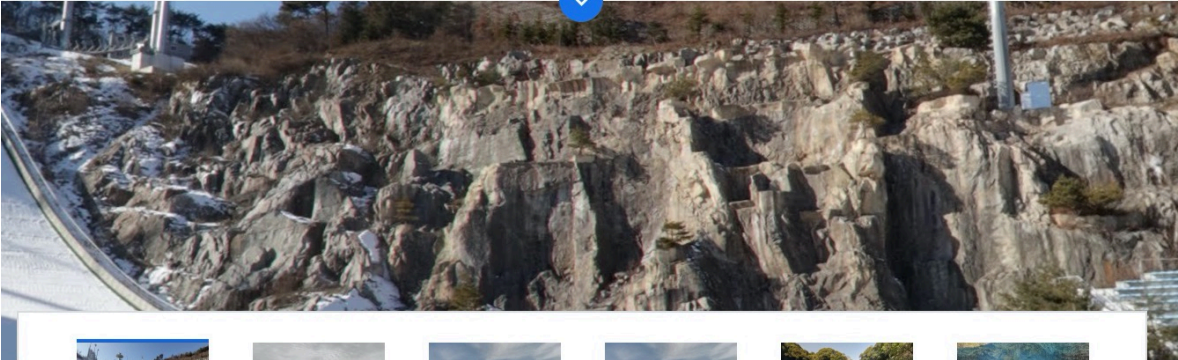



Google Street View

- <https://www.google.com/streetview/>


스트리트 뷰란 무엇인가요?

 Google 지도의 스트리트 뷰는 우리 주위의 환경을 수백만 개의 파노라마 이미지를 사용하여 Google 지도에 가상으로 표현한 것입니다. 스트리트 뷰 콘텐츠는 Google과 일반 참여자의 두 가지 출처를 통해 제공됩니다. 이와 같은 공동의 노력을 통해 전 세계의 사용자들이 가상으로 세계를 탐험할 수 있도록 돕고 있습니다.







강원도 하이라이트




대한민국의 국립공원




한국 하이라이트



서울 하이라이트




제주도



Ambrym Volcano,
Vanuatu

Google의 다음번 스트리트 뷰 촬영 예정지

Google에서 스트리트 뷰 차량 또는 스트리트 뷰 트레커로 촬영할 다음번 예정지가 어디인지 알아보세요.



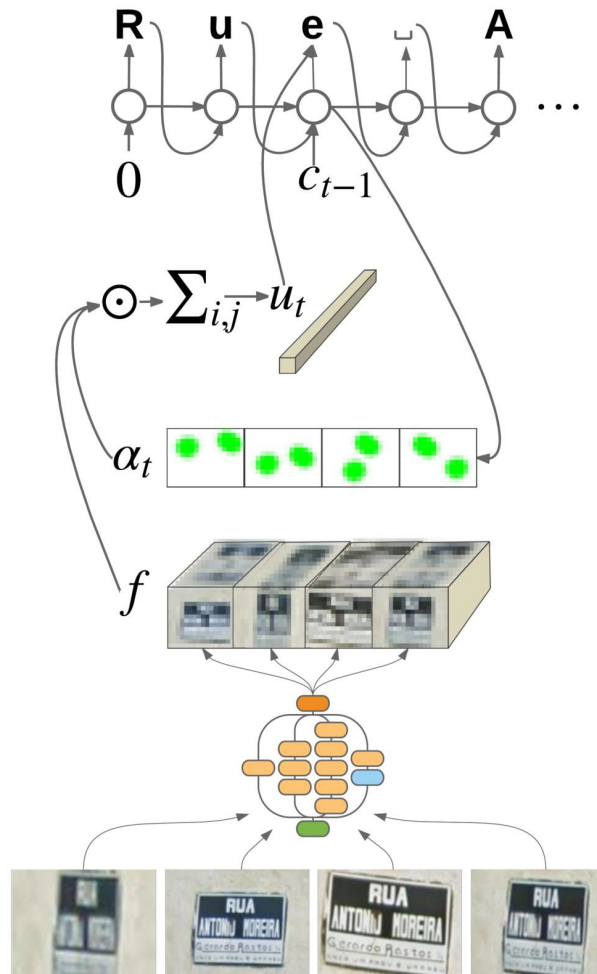
MAY 2021 /

United States

Alabama, Arizona, Florida, Georgia, ...[자세히 알아보기](#)

Methods

- CNN + Attention + RNN

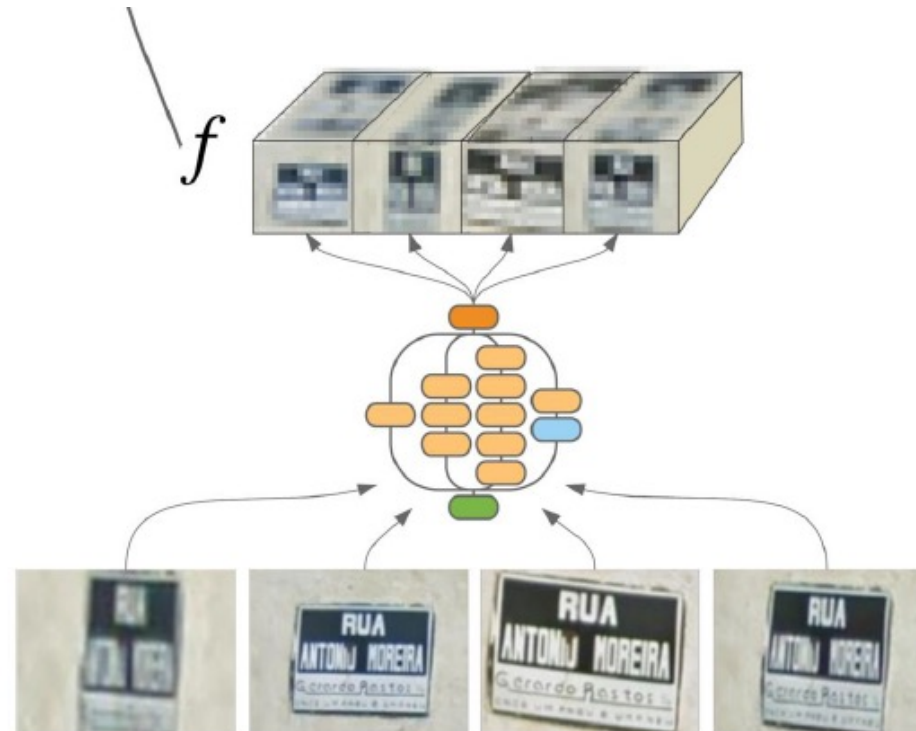


CNN-based feature extraction

- We consider 3 kinds of CNN: [inception-v2](#) [9], [inception-v3](#) [10] and [inception-resnet-v2](#) [10], which combines inception with resnets [12].
- We will use $f = \{f_{i,j,c}\}$ to denote the feature map derived by passing the image x through a CNN (here i, j index locations in the feature map, and c indexes channels).

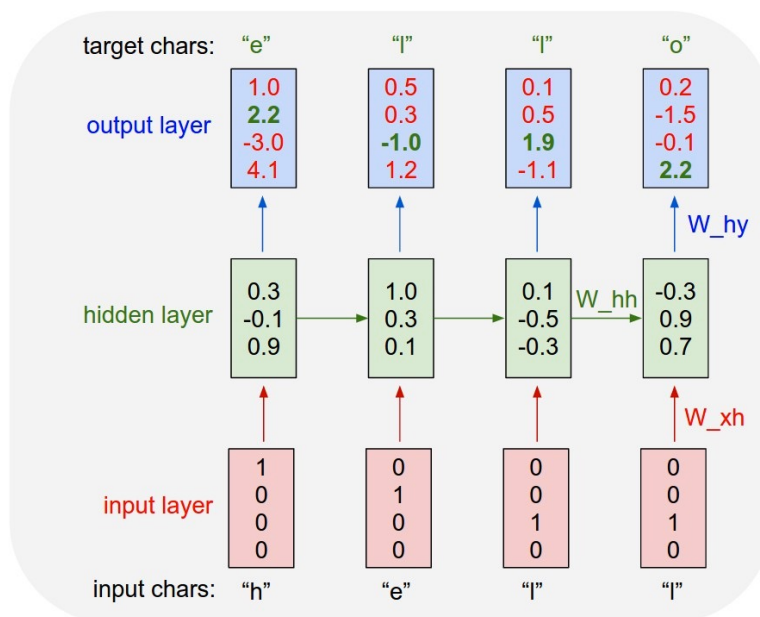
CNN-based feature extraction

- We pass each of the four views through the same CNN feature extractor, and then **concatenate the results into a single large feature map**, shown by the cube labeled “ f ”.

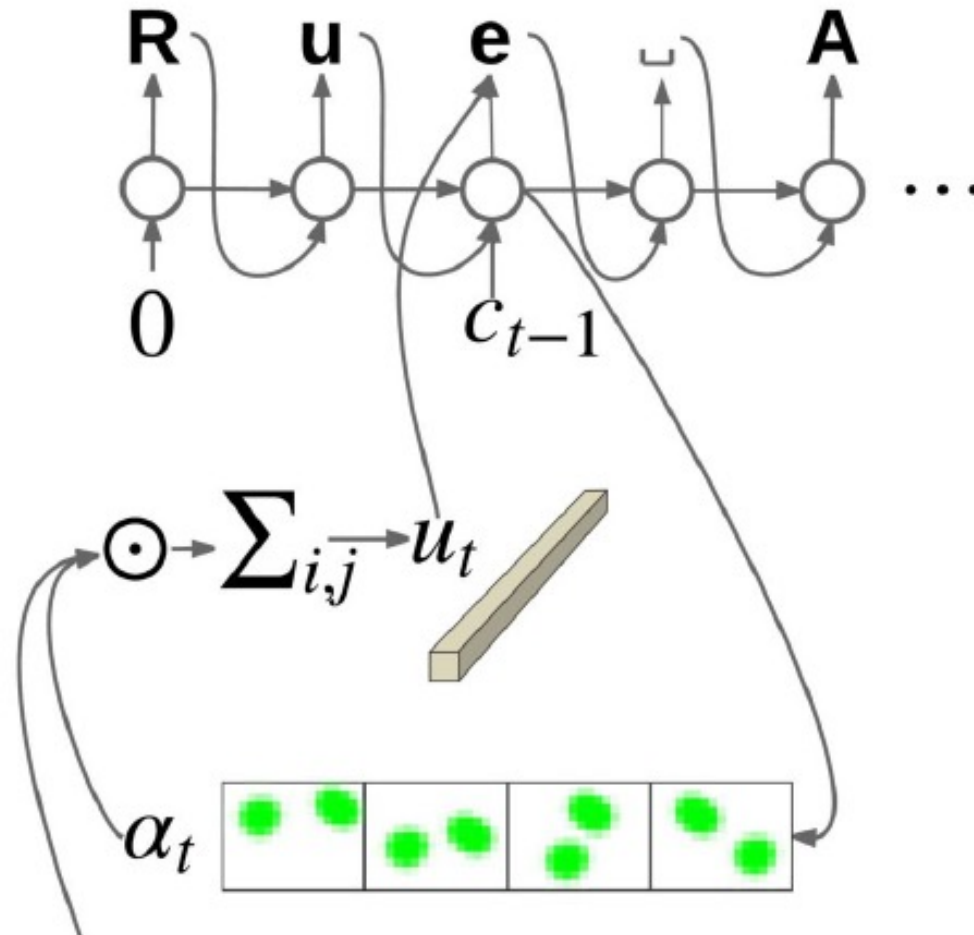


RNNs을 이용한 Character-level Language Modeling (Char-RNN)

- **Training** : 인풋 데이터(input data-x-)에서 글자(Character)를 하나 뒤로 민 타겟 데이터(target data-y-)로 RNNs을 학습한다.
- **Input Data** : 전체 문장 중 일정 길이의 글자들의 배열 (e.g. hell(전체 문장 중 일정 길이의 글자들의 배열))
- **Target Data** : 전체 데이터 중 Input Data를 한글자 뒤로 민 배열 (e.g. ello(전체 데이터 중 Input Data를 한글자 뒤로 민 배열))



RNN & Spatial Attention



RNN

- this acts as a **character level language model**, which takes inputs from the image, as we explain below.
- we compute a weighted combination of the features (the context) as follows:

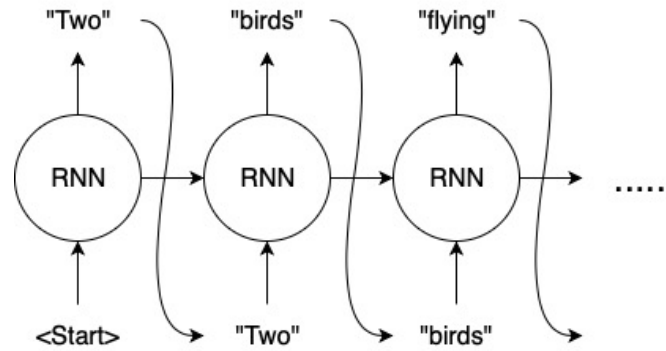
$$u_{t,c} = \sum_{i,j} \alpha_{t,i,j} f_{i,j,c}$$

- The total **input to the RNN** at time t is defined as

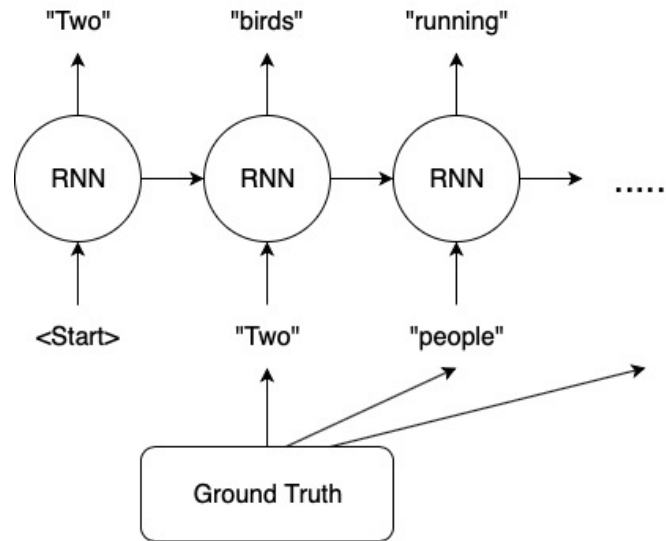
$$\hat{x}_t = W_c c_{t-1}^{OneHot} + W_{u_1} u_{t-1}$$

- where c_{t-1} is the index of the previous letter (**ground truth during training, predicted during test time**).

Teacher Forcing



Without Teacher Forcing



With Teacher Forcing

RNN

- We then compute the output and next state of the RNN as follows:

$$(o_t, s_t) = \text{RNNstep}(\hat{x}_t, s_{t-1})$$

- The final predicted distribution over letters at time t is given by

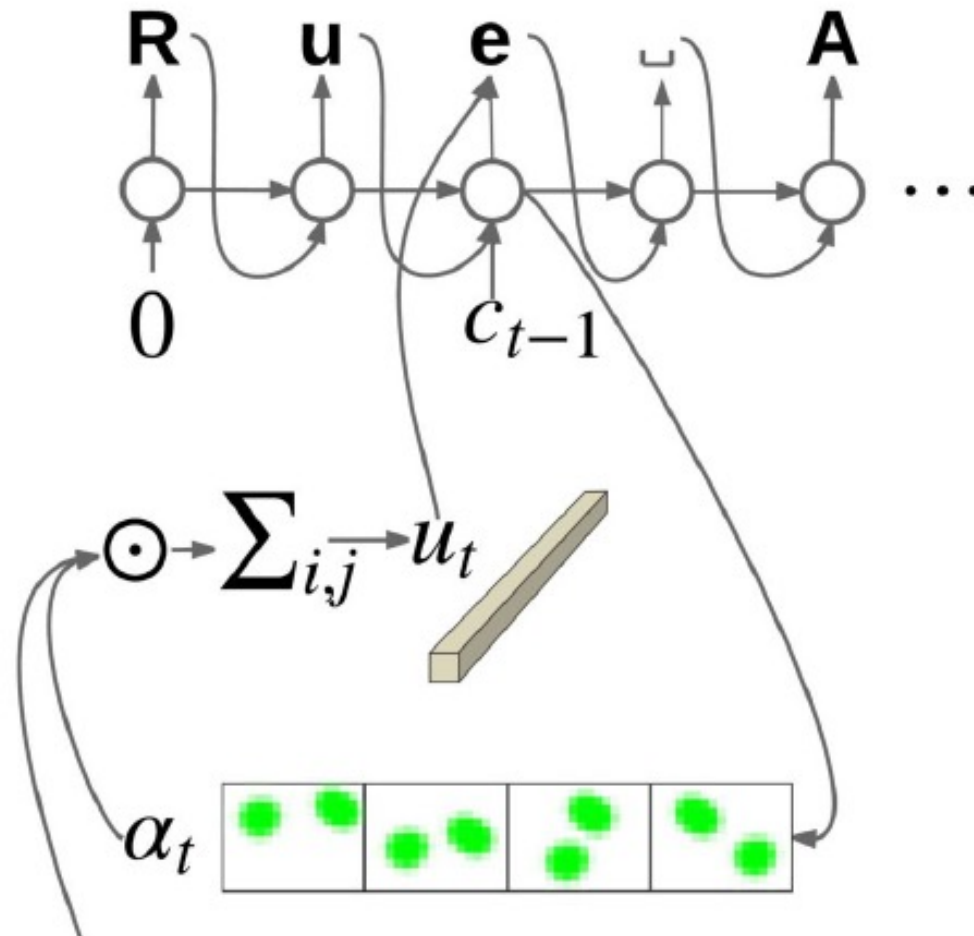
$$\hat{o}_t = \text{softmax}(W_o o_t + W_{u_2} u_t)$$

- This combines information from the RNN, o_t , with information from the attentional feature vector, u_t .
Finally, we compute the most likely letter:

$$c_t = \arg \max_c \hat{o}_t(c)$$

- This is called greedy decoding.

RNN & Spatial Attention



Spatial Attention

- Most prior works that use **spatial attention for OCR** (e.g., [1], [16]–[20]) predict the mask based on the current RNN state, as follows:

$$a_{t,i,j} = V_a^T \tanh(W_s s_t + W_f f_{i,j,:})$$

$$\alpha_t = \text{softmax}_{i,j}(a_t)$$

- where V_a is a vector and \tanh is applied elementwise to its vector argument. This combines content from the image, via $W_f f$, with a time-varying offset, via $W_s s_t$, to determine where to look.

Spatial Attention

- To make the model “location aware”, we concatenate $f_{i,j}$ with a one-hot encoding of the spatial coordinates (i, j) , as shown in Figure 2.
- More precisely, we replace the argument to the tanh function with the following:

$$W_s s_t + W_{f_1} f_{i,j,:} + W_{f_2} e_i + W_{f_3} e_j$$

- where e_i is a one-hot encoding of coordinate $f_{i,j}$, and similarly for e_j . This is equivalent to adding a spatially varying matrix of bias terms.

Spatial Attention

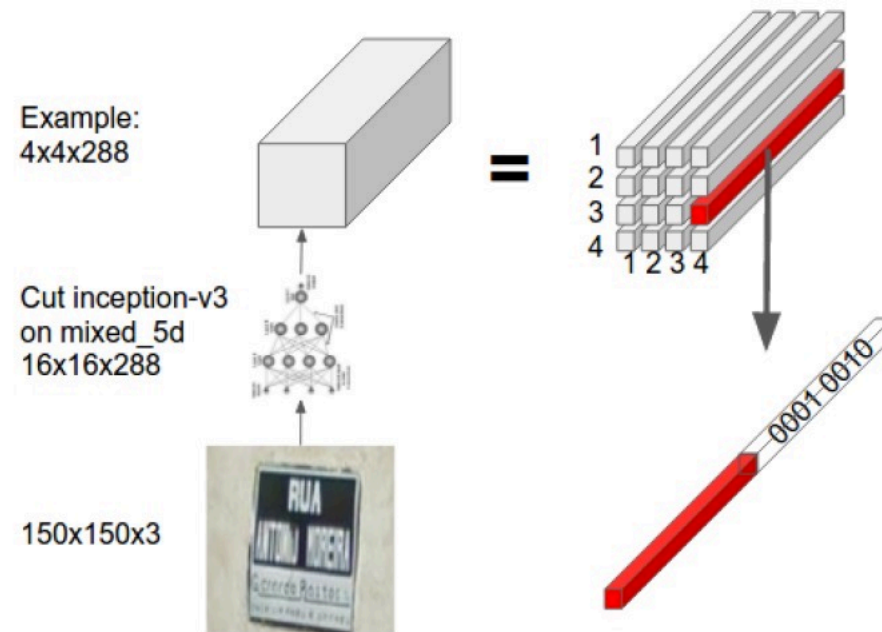


Fig. 2: Adding pixel coordinates to image features.

Handling Multiple View

- In the FSNS dataset, we have **four views for each input sign**, each of size **150x150**.
- We process each of these independently, through the same CNN-based feature extractor (parameters are shared), to compute four feature maps.
- We then **concatenate these horizontally** to create a single input feature map. For example, suppose the feature map for each of the four views is $16 \times 16 \times 320$; then after concatenation, the feature map $f_{i,j,c}$ will be **$64 \times 16 \times 320$** .



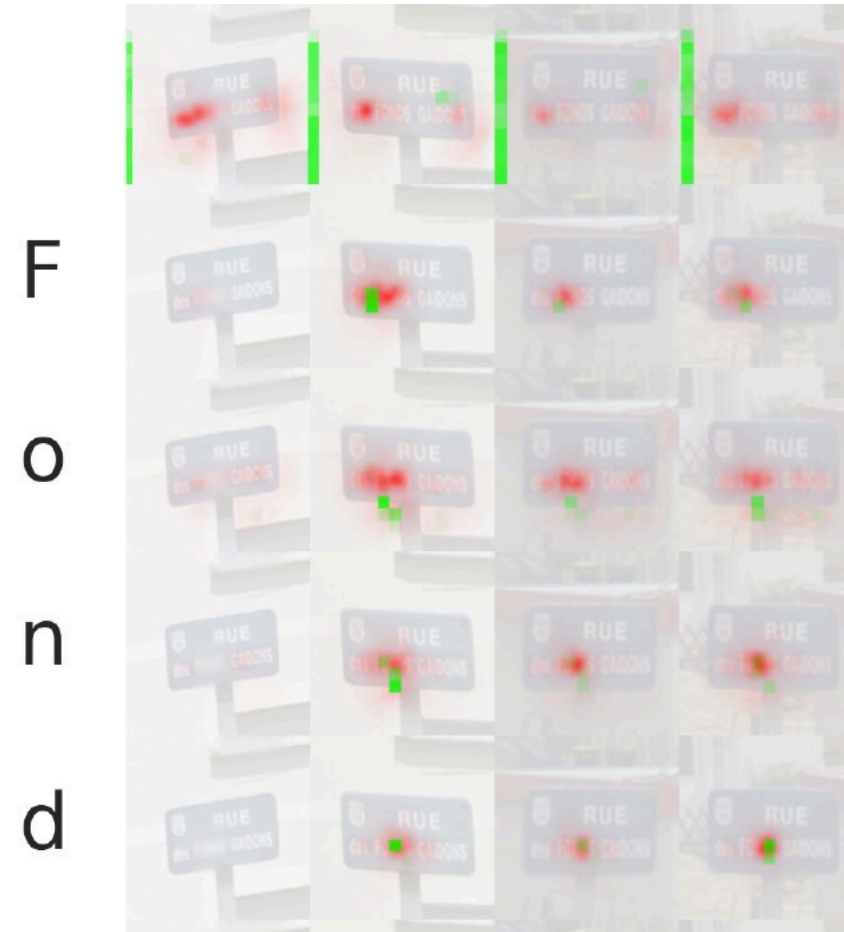
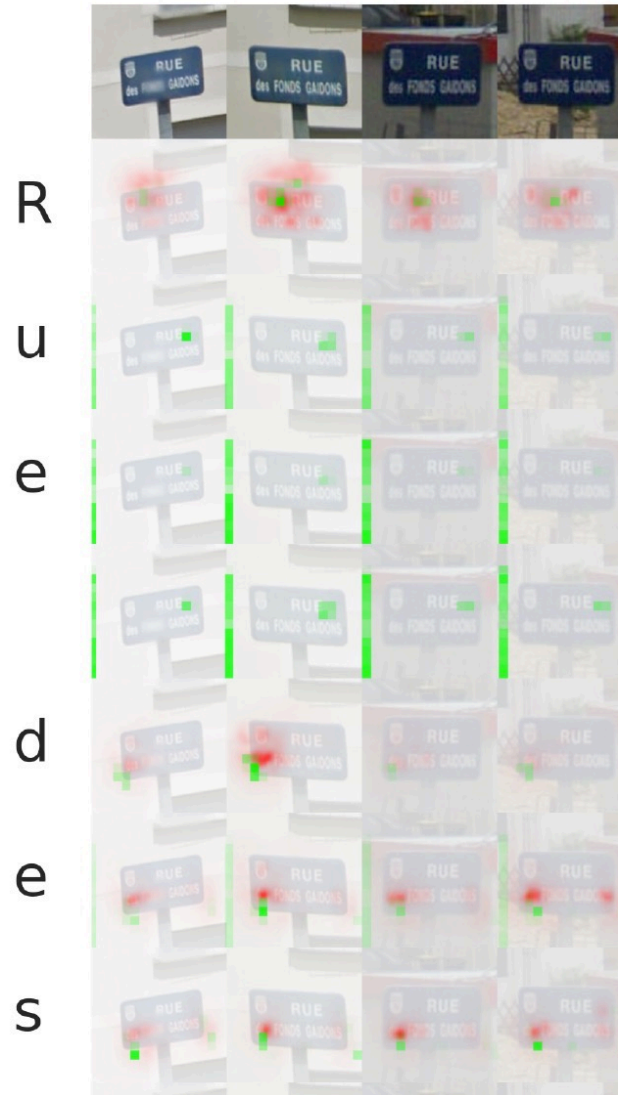
Handling Multiple View

- The FSNS dataset [7] contains 965,917 training images, 38,633 validation images and 42,839 test images.
- Each image has up to 4 tiles, intended to be a different view of the same physical street sign from Street View imagery from France. The size of every tile is 150x150 pixels.
- All the transcriptions of the street name are up to 37 characters long. (Our model takes advantage of this fact and always runs 37 steps, with an optional out-of-alphabet padded symbol.)
- There are 134 possible characters to choose from at each location, but most of the street names consist only of Latin letters.

Street View Business Names Dataset

- This is an internal dataset which contains ~ 1M single view images of business storefronts extracted from Google Street View imagery. See Figure 5 for some examples.
- The size of every image is 352x352. All transcriptions contain up to 33 symbols, with 128 characters in the vocabulary.

Visualization of saliency maps (in red) and attention masks (in green) on an FSNS image.



Visualization of the time-averaged saliency maps (in red) and attention masks (in green).



Accuracy

TABLE I: Accuracy on FSNS test set.

CNN	Attention	Accuracy
Smith et al. [7]	NA	72.46%
Inception-v2	Standard	80.7%
Inception-v2	Location	81.8%
Inception-v3	Standard	83.1%
Inception-v3	Location	84.0%
Inception-resnet-v2	Standard	83.3%
Inception-resnet-v2	Location	84.2%

Error Analysis

TABLE V: Breakdown of error types on FSNS.

Error type	Percent
Wrong ground truth	48
Wrong / Added / Missing accent over <i>e</i>	17
Wrong single letter inside the word	9
Wrong single letter at the beginning / end of word	8
Added / Missing hyphen (-)	7
Wrong full word	6
Read from the wrong view	3
Wrong / Added / Missing accent over different letter than e	2

Error Analysis



(a) Confused by font. Pred = 'Avenue Georges Frere', GT = 'Avenue General Frere'.



(b) Read text from the wrong view. Pred='Boulevard des Talus', GT='Boulevard Charles'.



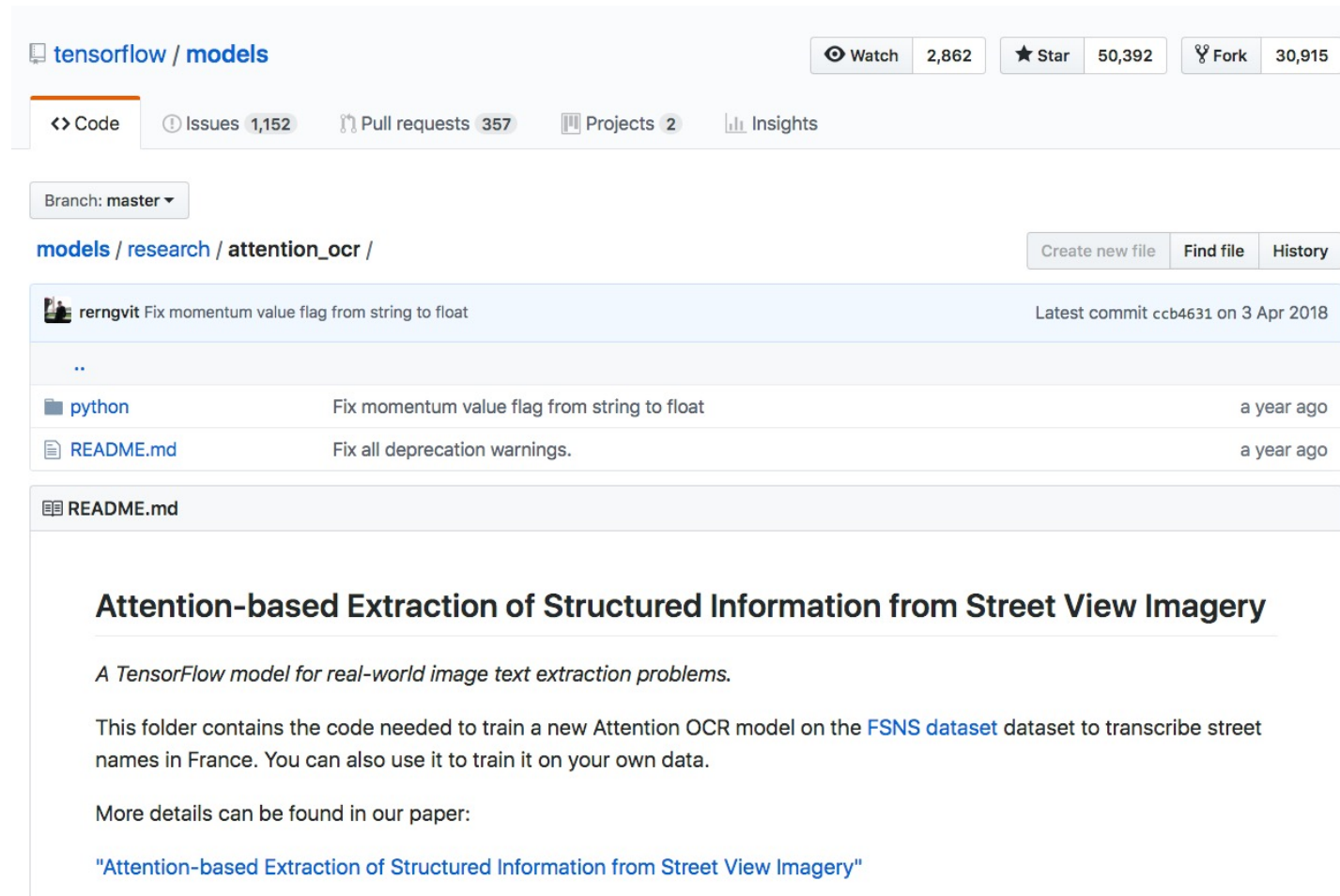
(c) Confusion due to scratched letter, which looks like 'J', but model uses its prior to produce 'O'. Pred='Impasse des Jorfèvres', GT='Impasse des Orfèvres'.



(d) The model has better language prior than the human annotator. Pred='Avenue des Erables', Wrong GT='Avenue des Enadles'.

Attention OCR Implementation

- https://github.com/tensorflow/models/tree/master/research/attention_ocr



The screenshot shows the GitHub repository page for `tensorflow / models`. The repository has 2,862 watches, 50,392 stars, and 30,915 forks. The navigation bar includes links for Code, Issues (1,152), Pull requests (357), Projects (2), and Insights. The current branch is `master`. The directory path is `models / research / attention_ocr /`. The commit history shows a recent commit by `rerngvit` titled "Fix momentum value flag from string to float" on 3 Apr 2018. The file list includes `python` (Fix momentum value flag from string to float, a year ago) and `README.md` (Fix all deprecation warnings, a year ago). The `README.md` file content is displayed below, featuring the title "Attention-based Extraction of Structured Information from Street View Imagery" and a description of the TensorFlow model for real-world image text extraction problems.

Attention-based Extraction of Structured Information from Street View Imagery

A TensorFlow model for real-world image text extraction problems.

This folder contains the code needed to train a new Attention OCR model on the [FSNS dataset](#) dataset to transcribe street names in France. You can also use it to train it on your own data.

More details can be found in our paper:

["Attention-based Extraction of Structured Information from Street View Imagery"](#)

Thank you!
