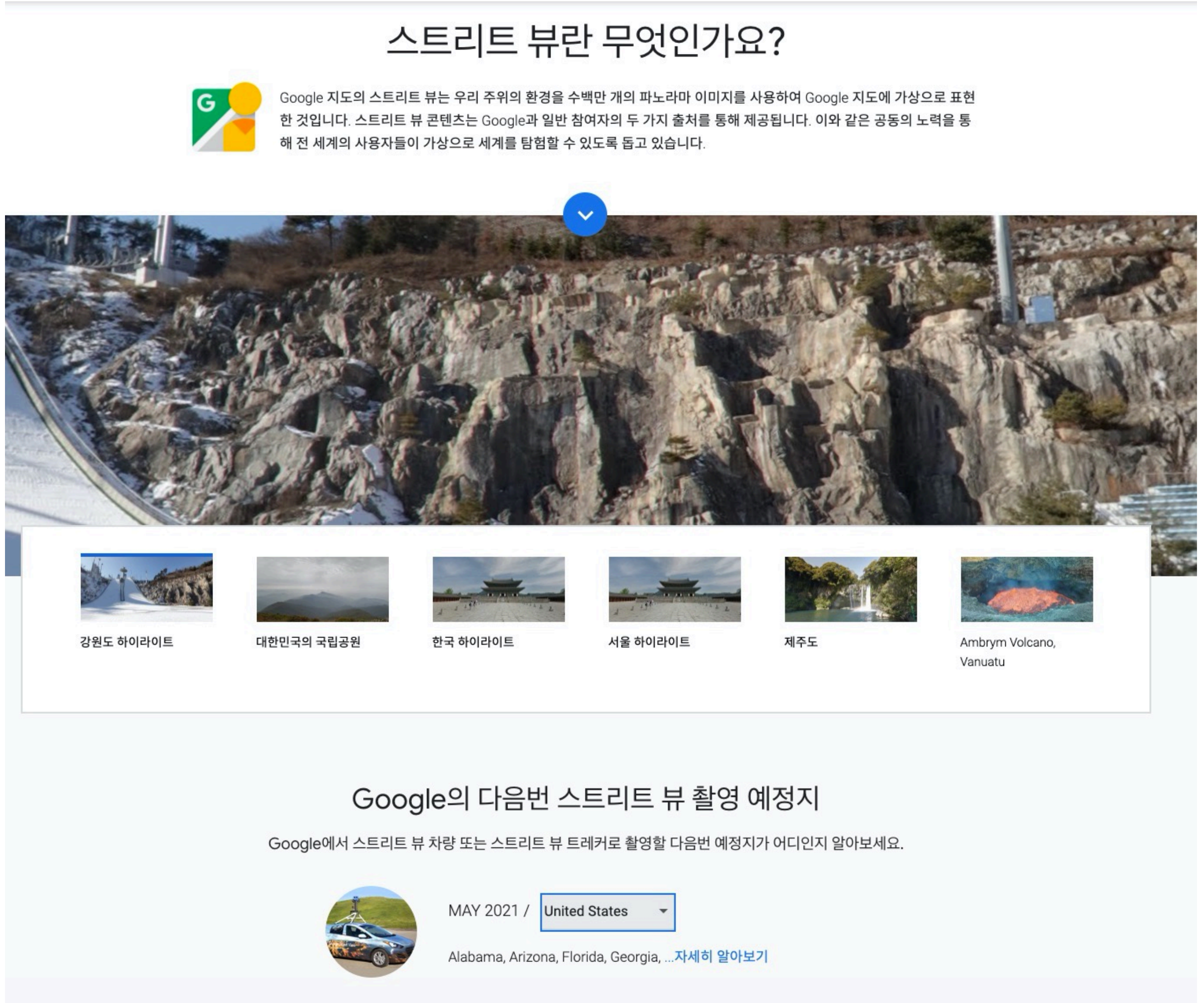
Attention OCR

- 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.

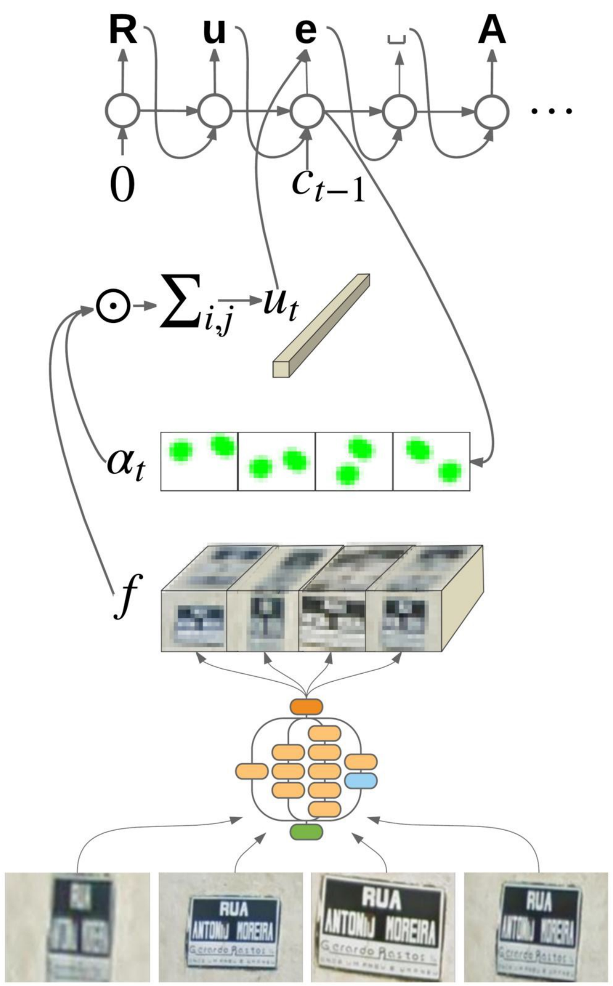
Google Street View

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



Methods

- CNN + Attention + RNN



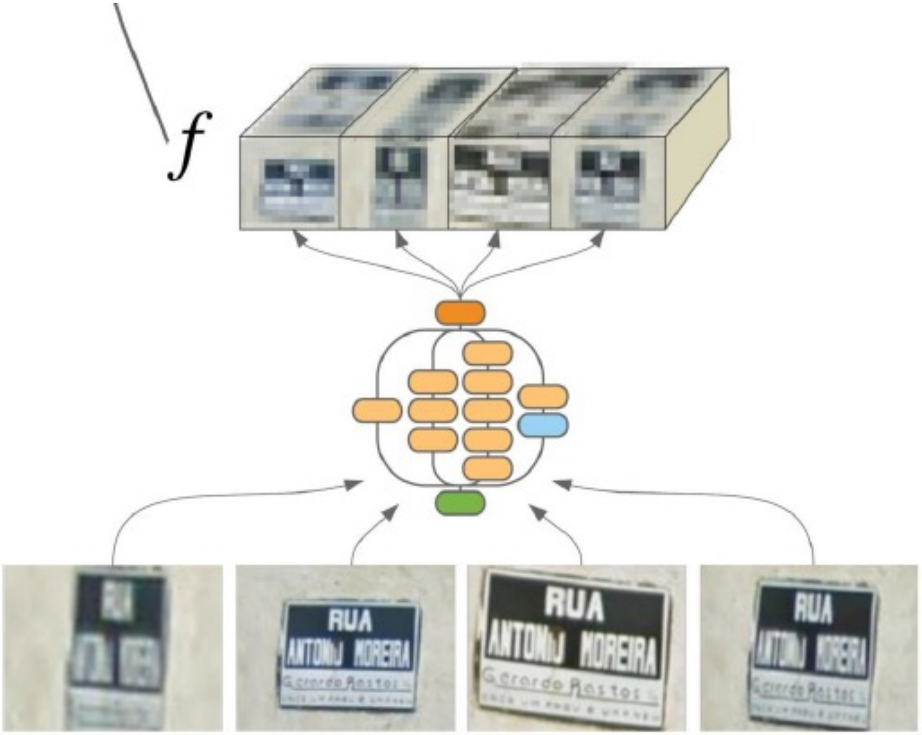
AI School.

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

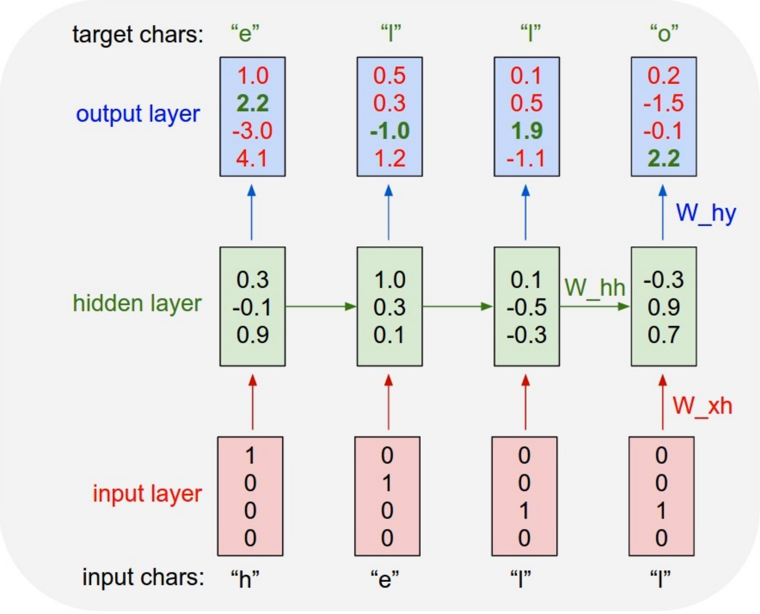
- We pass each of the four views through the same CNN feature extractor, and then concatenate the results into a single large feature map, shown by the cube labeled "f".



AI School.

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

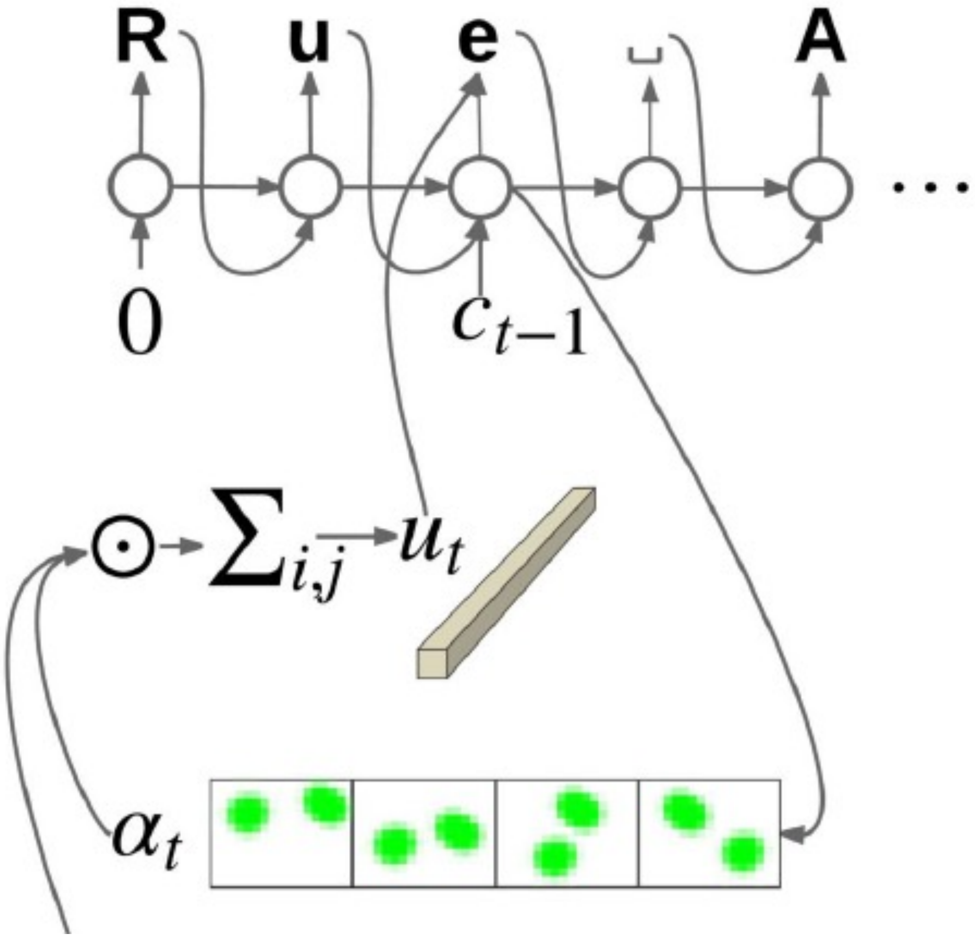
- Training :인풋 데이터(input data-x-)에서 글자(Character)를 하나 뒤로 민 타겟 데이터(target data-y-)로 RNNs을 학습한다.
- Input Data : 전체 문장 중 일정 길이의 글자들의 배열 (e.g. hell(전체 문장 중 일정 길이의 글자들의 배열))
- Target Data : 전체 데이터중 Input Data를 한글자 뒤로 민 배열 (e.g. ello(전체 데이터중 Input Data를 한글자 뒤로 민 배열)



Reference : http://karpathy.github.io/2015/05/21/rnn-effectiveness/

AI School.

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:

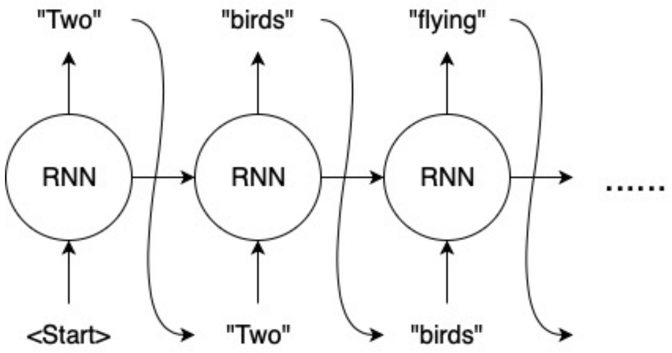
$$u_{t,c} = \sum_{i,j} \alpha_{t,i,j} f_{i,j,c}$$

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

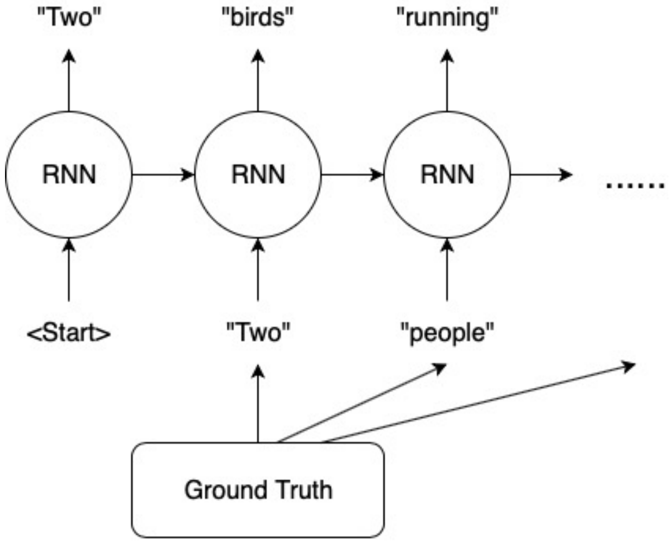
$$\hat{x}_t = W_c c_{t-1}^{OneHot} + W_{u_1} u_{t-1}$$

- 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

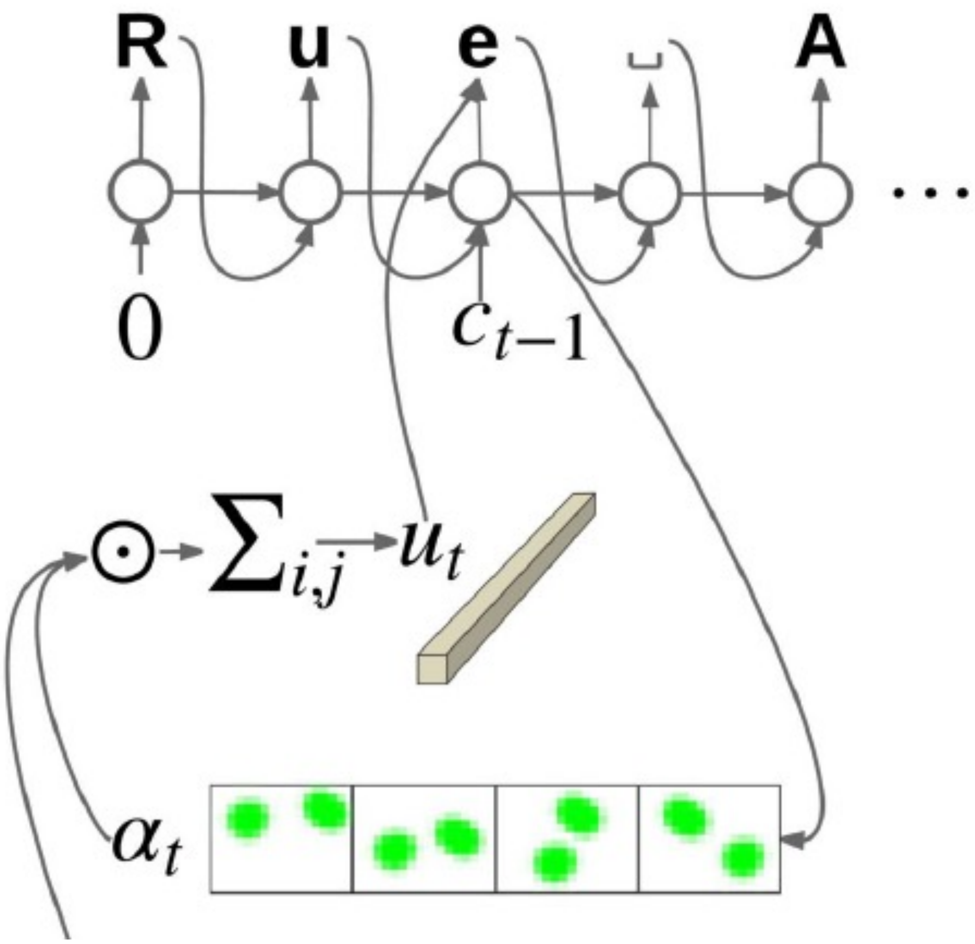
$$\hat{o}_t = \text{softmax}(W_o o_t + W_{u_2} u_t)$$

- 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.

AI School.

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

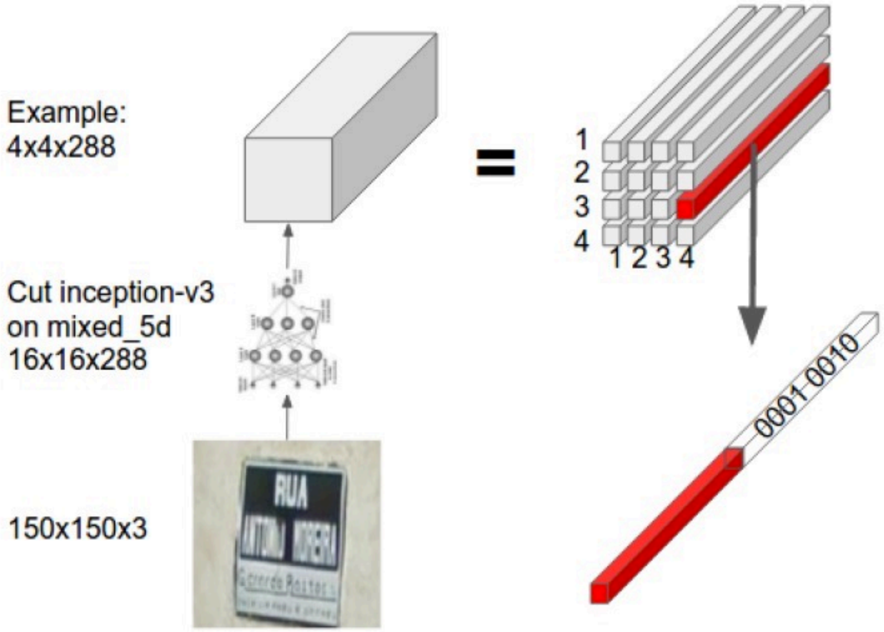


Fig. 2: Adding pixel coordinates to image features.

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 x 16 x 320; then after concatenation, the feature map $f_{i,j,c}$ will be 64 x 16 x 320.

AI School.

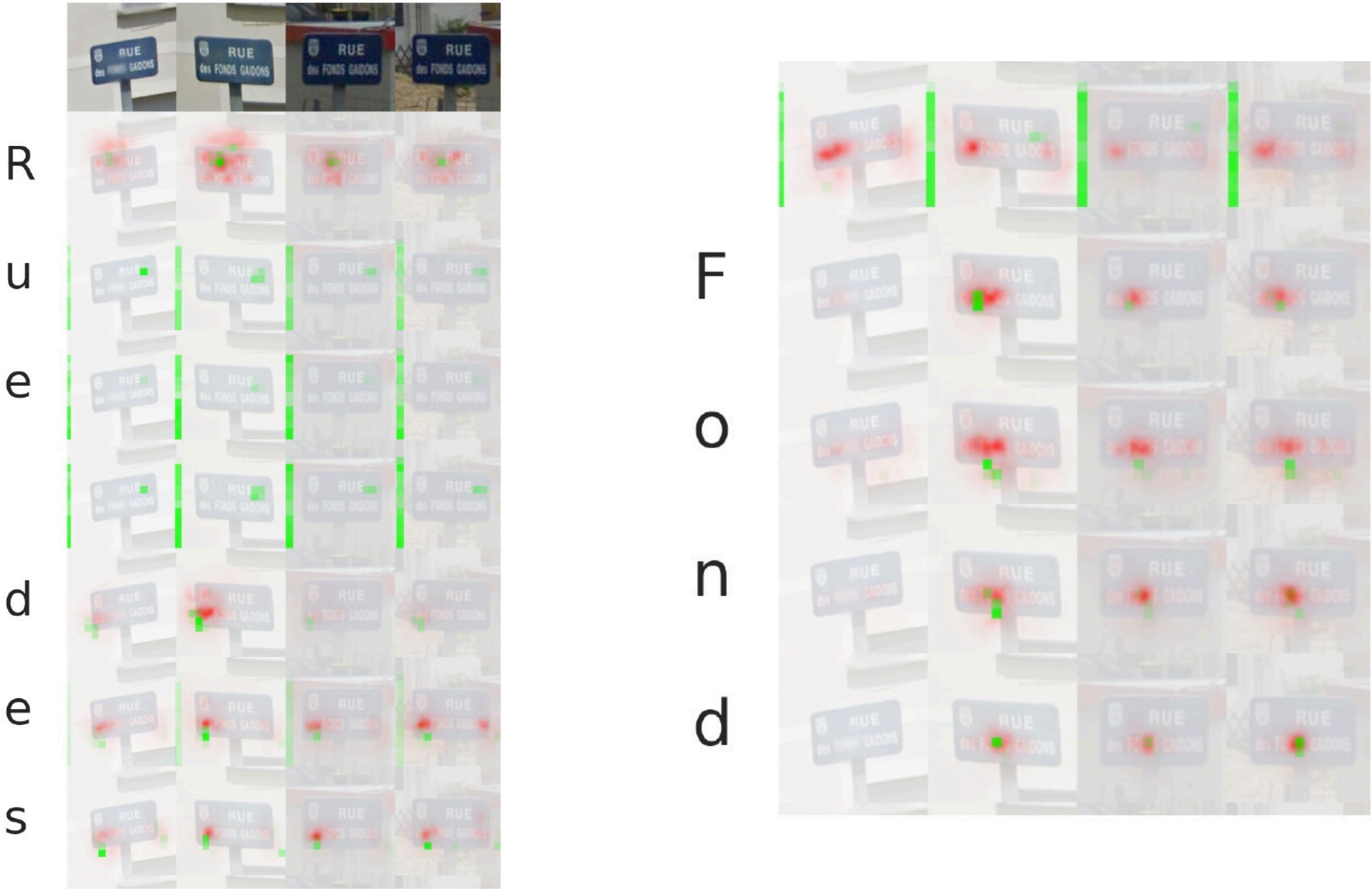
Handling Multiple View

- 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.



AI School.

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

Accuracy

TABLE I: Accuracy on FSNS test set.

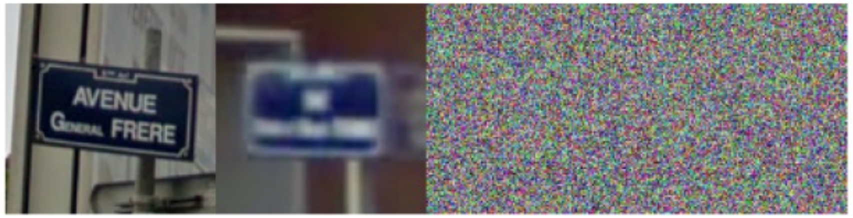
CNN	Attention	Accuracy
Smith et al. [7]	NA	72.46%
Inception-v2	Standard	80.7%
Inception-v2	Location	81.8%
Inception-v3	Standard	83.1%
Inception-v3	Location	84.0%
Inception-resnet-v2	Standard	83.3%
Inception-resnet-v2	Location	**84.2%**

Error Analysis

TABLE V: Breakdown of error types on FSNS.

Error type	Percent
Wrong ground truth	48
Wrong / Added / Missing accent over *e*	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

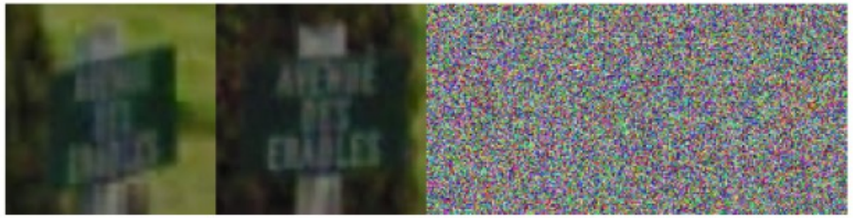
Error Analysis



(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'.

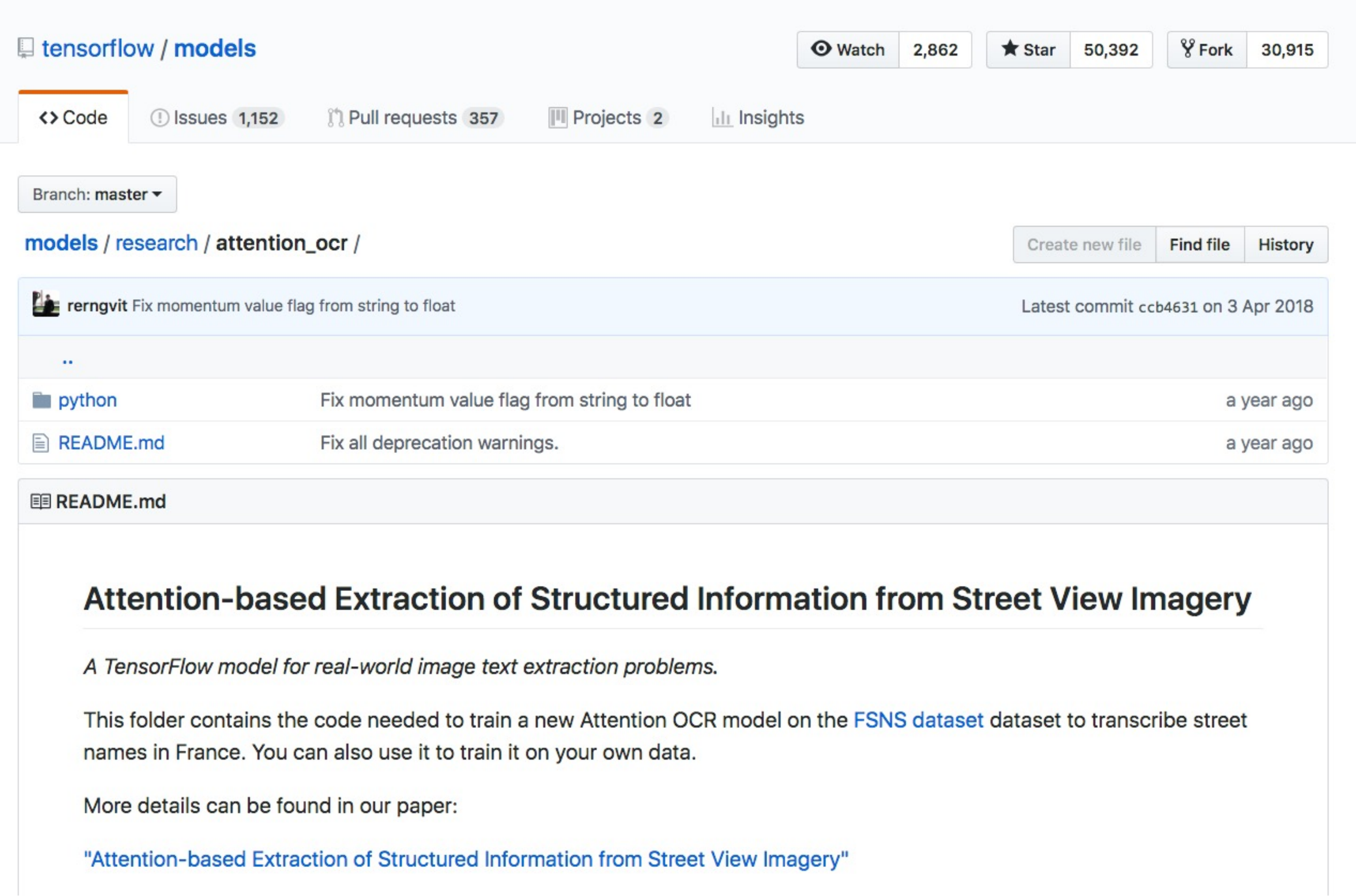
(c) Confusion due to scratched letter, which looks like 'J', but model uses its prior to produce 'O'. Pred='Impasse des Jorfèvres', GT='Impasse des Orfévres'.



(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



Thank you!