Being able to automatically describe the content of an image using properly formed English sentences is a challenging task, but it could have a great impact by helping visually impaired people better understand their surroundings.

These images can then be used to generate captions that can be read out loud to the visually impaired so that they can get a better sense of what is happening around them.

Two people walking on a street

Description automatically generated with low confidence

Challenges that blind People Face

**Technical Approach to Solve Problem**

We implemented a deep recurrent architecture that automatically produces a short description of images. Our models use a CNN, which was pre-trained on ImageNet, to obtain images features. We then feed these features into either a vanilla RNN or a LSTM network (Figure 2) to generate a description of the image in valid English language.

**CNN-based Image Feature Extractor**

For feature extraction, we use a CNN. CNNs have been widely used and studied for images tasks, and are currently state-of-the-art methods for object recognition and detection. Concretely, for all input images, we extract features from the fc7 layer of the VGG-16 network pre-trained on ImageNet which is very well tuned for object detection. We obtained a 4096-Dimensional image feature vector that we reduce using Principal Component Analysis (PCA) to a 512-Dimensional image feature vector due to computational constraints. We feed these features into the first layer of our RNN or LSTM at the first iteration.

**RNN-based Sentence Generator**

We first experiment with vanilla RNNs as they have been shown to be powerful models for processing sequential data [25, 26]. Vanilla RNNs can learn complex temporal dynamics by mapping input sequences to a sequence of hidden states, and hidden states to outputs via given equations.

A picture containing text, antenna

Description automatically generated

where f is an element-wise non-linearity, ht 2 RN is the hidden state with N hidden units, and yt is the output at time t. In our implementation, we use a hyperbolic tangent as our element-wise nonlinearity. For a length T input sequence x1; x2; :::; xT , the updates above are computed sequentially as h1 (letting h0 = 0), y1, h2, y2,… hT , yT .

Diagram

Description automatically generated

Figure 2: Image Retrieval System and Language Generating Pipeline.

**LSTM-based Sentence Generator**

Although RNNs have proven successful on tasks such as text generation and speech recognition [25, 26], it is difficult to train them to learn long-term dynamics. This problem is likely due to the vanishing and exploding gradients problem that can result from propagating the gradients down through the many layers of the recurrent networks. LSTM networks (Figure 3) provide a solution by incorporating memory units that allow the networks to learn when to forget previous hidden states and when to update hidden states when given new information.

**Training**

We train our LSTM model to correctly predict the next word (yt) based on the current word (xt), and the previous context (ht􀀀1). We do this as follows: we set h0 = 0, x1 to the START vector, and the desired label y1 as the first word in the sequence. We then set x2 to the word vector corresponding to the first word generated by the network. Based on this first word vector and the previous context the network then predicts the second word, etc. The word vectors are generated using the word2vec embedding model as described by Mikolov et. al [1]. During the last step, xT represent the last word, and yT is set to an END token.

**Testing**

To predict a sentence, we obtain the image features bv, set h0 = 0, set x1 to the START vector, and compute the distribution over the first word y1. Accordingly, we pick the argmax from the distribution, set its embedding vector as x2, and repeat the procedure until the END token is generated.

Diagram, schematic

Description automatically generated

Figure 3: LSTM unit and its gates

**Optimization**

We use Stochastic Gradient Descent (SGD) with mini-batches of 25 imagesentence pairs and a momentum of 0.95. We cross-validate the learning rate and the weight decay. We achieved our best results using Adam, which is a method for efficient stochastic optimization that only requires first-order gradients and computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. Adam’s main advantages are that the magnitudes of parameter updates are invariant to rescaling of the gradients, its step-size is approximately bounded by the step-size hyperparameter, and it automatically performs a form of step-size annealing.

**Dataset**

For this exercise, we will use the 2014 release of the Microsoft COCO dataset which has become the standard testbed for image captioning. The dataset consists of 80,000 training images and 40,000 validation images, each annotated with 5 captions written by workers on Amazon Mechanical Turk. Four example images with captions can be seen in Figure 4. We convert all sentences to lowercase and discard non-alphanumeric characters.

A collage of a person playing baseball

Description automatically generated with low confidence

Figure 4: Example images and captions from the Microsoft COCO Caption dataset.

**Qualitative Results**

Our models generates sensible descriptions of images in valid English (Figure 6 and 7). As can be seen from example groundings in Figure 5, the model discovers interpretable visual-semantic correspondences, even for relatively small objects such as the phones in Figure 7. The generated descriptions are accurate enough to be helpful for visually impaired people. In general, we find that a relatively large portion of generated sentences (60%) can be found in the training data.

Table

Description automatically generated

Figure 5: Evaluation of full image predictions on 1,000 test images of the Microsoft COCO 2014 dataset

A group of people walking down a sidewalk with umbrellas

Description automatically generated with low confidence

Figure 6: Example image descriptions generated using the RNN structure.

A picture containing text, outdoor, building, road

Description automatically generated

Figure 7: Example image descriptions generated using the LSTM structure.

**Conclusion**

We created a deep learning model that automatically generates image captions with the goal to helping visually impaired people to better understand their surroundings environments.

**References**

[*https://cs224d.stanford.edu › reports › mcelamri*](https://cs224d.stanford.edu/reports/mcelamri.pdf)

<https://www.semanticscholar.org/paper/Automated-Neural-Image-Caption-Generator-for-People-Elamri-Planque/cc61cd90529fede6e1bfb14042d021bc2a076e99>

[*https://www.livestrong.com/article/241936-challenges-that-blind-people-face*](https://www.livestrong.com/article/241936-challenges-that-blind-people-face/)

[1] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. ”Distributed representations of words and phrases and their compositionality.” Advances in Neural Information Processing Systems (NIPS)26: 3111–3119 (2013). Web. 29 Apr. 2016