Independent Study Report

Xiaodan Du

Fall 2018

1. Background

Although scene understanding systems have made a huge progress in recent years and can do very well on common images, the “unusual” images, which may contain uncommon scenes and combinations of objects often break a lot of scene understanding systems.

The goal of this study is to implement a classifier to sort out “unusual” images from the majority “usual” ones. As its name implies, “unusual” images are very rare in the widely used datasets such as Microsoft’s COCO (Common Objects in Context) and Stanford’s ImageNet. Therefore, picking out “unusual” images by humans will cost a large amount of time and money, not mentioning potential bias caused by different human labelers. Training a binary classifier could help us to significantly enhance the accuracy and efficiency in selecting out “unusual” images. With the help of a high accuracy classifier, we can even create a dataset contains only unusual images. With the help of it, researchers can better understand the properties of “unusual” images and develop more advanced scene understanding systems to tackle the problem.

In this report I will present:

1. An introduction of the dataset created by student Sai Krishna Bollam
2. An image classifier trained on the dataset and its performance
3. A text classifier trained on captions corresponding to images in the dataset
4. How the two classifiers are combined to perform ensemble classification
5. Evaluation and analysis of the results
6. Dataset
   1. Introduction of the dataset

The dataset I use is mainly the work of Sai Krishna Bollam. A corpus of 10000 images is randomly selected from the MS COCO dataset and uploaded to Amazon Mechanical Turk for crowdsourcing. Turkers are asked to label the given images on a scale of 1 to 3 based on their judgement of the unusualness of the image. The unusualness scale and its corresponding meaning is listed in Table 1.

|  |  |
| --- | --- |
| Unusualness scale | Instructions |
| 1 | Usual |
| 2 | Somewhat unusual |
| 3 | Very unusual |

Table 1 Unusualness scale and its corresponding instructions

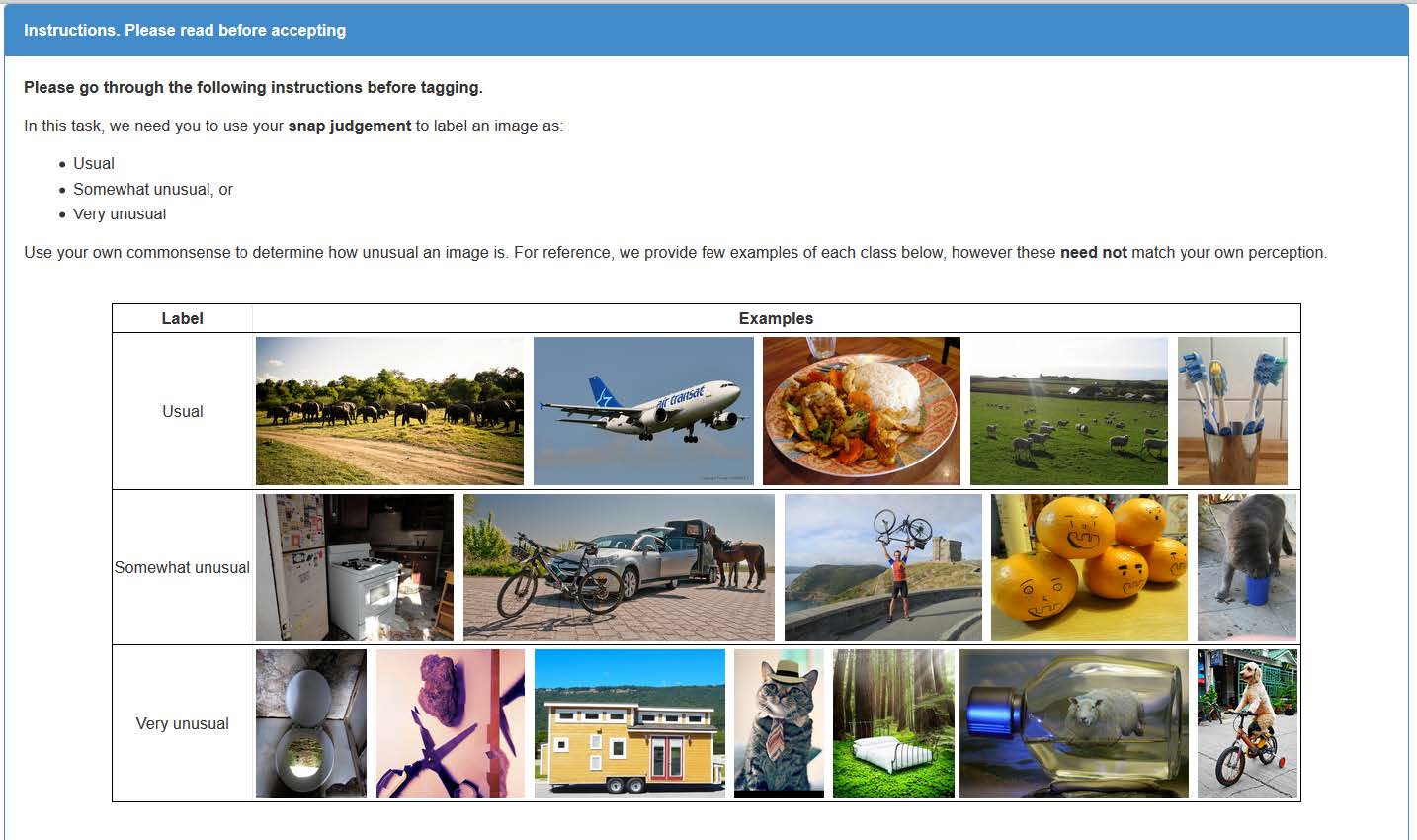


Figure 1 Instructions for crowdsourcing images on Amazon Mechanical Turk

Each image is labeled by three or more Turkers. The final unusualness score for an image is the average of all scores it received. Percentage distribution of average rating is listed in Table2. Images with unusualness score equal or higher than 2.0 are classified as “unusual” and those with 1.0 are classified as “usual”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classification | Avg. rating | No. of images | Percentage | Accumulative Percentage |
| **Unusual: 310** | 3.00 | 2 | 0.02% | 0.02% |
| 2.83 | 4 | 0.04% | 0.06% |
| 2.67 | 13 | 0.13% | 0.19% |
| 2.50 | 20 | 0.20% | 0.39% |
| 2.33 | 31 | 0.31% | 0.70% |
| 2.17 | 65 | 0.65% | 1.35% |
| 2.00 | 175 | 1.75% | 3.10% |
| Not included in the current dataset | 1.83 | 166 | 1.66% | 4.76% |
| 1.67 | 789 | 7.89% | 12.65% |
| 1.50 | 270 | 2.70% | 15.35% |
| 1.33 | 2442 | 24.42% | 39.77% |
| 1.17 | 464 | 4.64% | 44.41% |
| **Unusual: 5559** | 1.00 | 5559 | 55.59% | 100.00% |

Table 2 Score distribution obtained from Amazon Mechanical Turk

Because it is the “unusual” class that we are interested in, percentage of “unusual” images in the testing set is set to 25%, much higher than that in the training set. Partitioning of the dataset can be found in Table 3.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train | Test | Total |
| Unusual | 210 | 100 | 310 |
| Usual | 5259 | 300 | 5559 |
| Total | 5469 | 400 | 5869 |

Table 3 Dataset partitions

* 1. Problems of the dataset

From Table 2 one can easily tell the resulting dataset is a highly imbalanced dataset. The ratio of unusual to usual is approximately 1: 18. As a result, weighted loss and sampling tricks are used in model training. Details will be provided later in the report.

It is also worth mentioning that due to ambiguity of the definition of “unusualness” and the relatively small number of labels for each image, biases caused by Turkers’ subjective perspective have a significant influence on the dataset.



(a)



(b)



(c)



(d)

Figure Several examples of unusual images from the dataset

Figure 2(a) has a score of 2.00 therefore is classified as “unusual”. The reason for it to be considered “unusual” is probably the turkers don’t speak Arabic. Currently Amazon Mechanical Turk is only available in English and requires a U.S. address to register. Due to limited cultural backgrounds of the turkers, images considered unusual by a group of people might be considered completely common by the rest of the world. Besides turkers’ personal bias, descriptions of unusualness given in Turk interface is too broad and vague.

Figure 2(b) is also received an average score of 2.00. Bananas on a magazine is by no means unusual. Most of the images in the dataset are labelled by no more than six turkers. As a result, the score of an image will be significantly affected if any of the turkers are biased. Also, directly asking people whether they think an image is unusual influences their judgement by the psychological hint that there might be something special of the image. This tendency introduces a shift of score to the higher end. The problem is very common in the dataset.

Figure 2(c) and Figure 2(d) have score 2.33 and 3.00, respectively. These two are the examples of what I think “unusual” really means. Unfortunately, these kinds of data is rare in the dataset. These images are probably considered noise by the network because of the rarity.

1. Image classifier

A pre-trained ResNet50 network is used as a fixed feature extractor for my binary image classifier. The network’s last layer was replaced by a fully connected layer with two output nodes so that it can generate a score for each class. When trained with random sampling, though the overall accuracy is very high, accuracy for “unusual” images is very low. With an excessive amount of “usual” images in the training set, the network tends to blindly classify every data as “usual”. To solve the problem, a weighted sampler is used to perform balanced sampling between the two classes. Then several loss functions are tested and cross entropy function is selected based on performance.

Since the dataset is highly imbalanced and we are only interested in sorting out the “unusual” ones, overall accuracy is not helpful in this case. Instead, receiver operating characteristic curve (ROC) curve and area under the curve (AUC) analysis is used to measure the performance of a model.

With the combination of hyper-parameters listed in Table 4, the binary image classifier can achieve an AUC value of 0.80 for “unusual” class.

|  |  |
| --- | --- |
| Hyper-parameters | Values |
| Epoch | 20 |
| Learning rate | 0.0005 |
| Momentum | 0.9 |
| Learning rate decay | 0.1/7 epochs |
| Loss function | Cross Entropy Loss |
| Batch size | 64 |

Table 4 Hyper-parameter selection for the image classifier

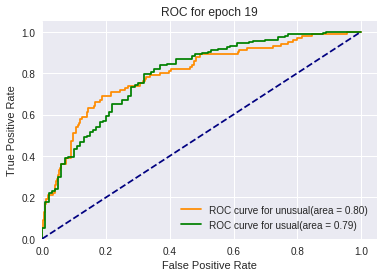


Figure 3 AUC and ROC curves for the image classifier

1. Text classifier

Each image in COCO dataset comes with four or five captions. The intention of training a text classifier on image annotations is to see if captions contain linguistic cues that are specific properties of unusualness.

1. Baseline

To measure the performance of the text classifier, a baseline is implemented. First, each word is transformed into a vector of a dimension of 50 using pre-trained GloVe vocabulary. Then vectors for words within a caption are averaged. The averaged vector is then used as an input for a simple 3-layer-deep fully connected network. The baseline achieves a 0.73 AUC for “unusual” and a 0.72 AUC for “usual”.

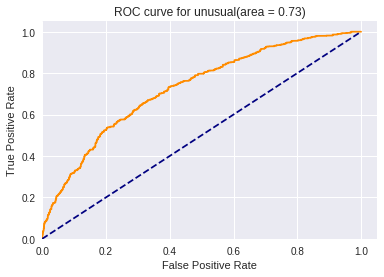


Figure 4 Baseline AUC for "unusual" class

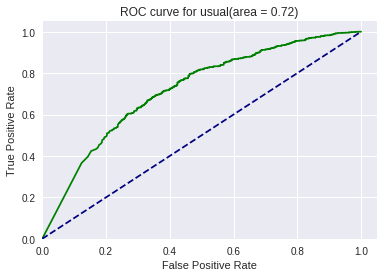


Figure 5 Baseline AUC for "usual" class

1. RNN classifier

An GRU network is constructed to classify captions of images in the dataset. Captions from both classes are randomly selected with equal probability from the dataset and fed to a GRU network. A weighted loss with a 2 : 1 ratio (unusual : usual) is also applied to further boost the accuracy for “unusual” data. Two different data preprocessing schemes are tested: 1. all captions of an image are combined and embedded into one long vector, 2. all captions are embedded separately into independent vectors. The first scheme results in an AUC for “unusual” around 0.67, lower than that of the second scheme. Therefore the second scheme is adopted for constructing the GRU network.

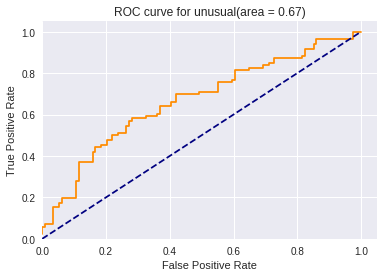


Figure 6 AUC and ROC curves of “unusual” for Scheme 1

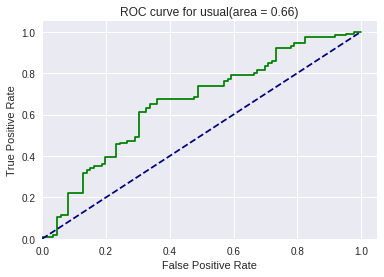


Figure 7 AUC and ROC curves of “usual” for Scheme 1

The RNN network contains three recurrent layers with 100 features for each hidden state. Cross Entropy loss is used as criterion for the text classifier. The GRU network achieves an AUC score of 0.76 for “unusual”, slightly better than the baseline.

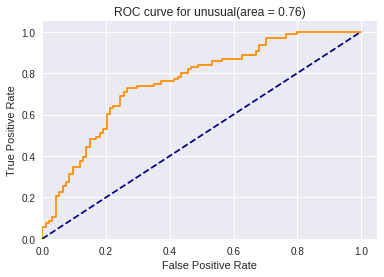


Figure 8 AUC and ROC curves of "unusual" for the text classifier

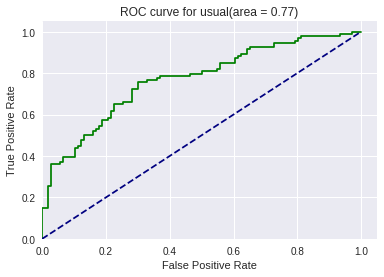


Figure 9 AUC and ROC curves of "usual" for the text classifier

The hyper-parameter selection is given in Table 5.

|  |  |
| --- | --- |
| Hyper-parameters | Values |
| Epoch | 200 |
| Learning rate | 0.01 |
| Recurrent layer | 3 |
| Hidden feature | 100 |
| Loss function | Cross Entropy Loss |
| Batch size | 100 |

Figure 10 Hyper-parameter selection for the text classifier

1. Combined Classification
   1. Architecture

Classification based on the combination of image features and text features

With both classifiers developed separately, we want to see if the combination of visual and textual information may enhance the performance. Each image loaded is fed into the image classifier. The image’s corresponding caption is fed into the text classifier. Each classifier outputs a 2 by 1 vector which are mapped to [0,1] by Sigmoid function. The class with higher addition is selected to be the prediction. Figure 11 is a visual representation of the predicting process.

Figure Architecture for the combined classifier

Image

Caption

Cross Entropy

Cross Entropy

Image Classifier

Text Classifier

2x1 Score vector

2x1 Score vector

Sigmoid

Sigmoid

2x1 0~1 vector

2x1 0~1 vector

2x1 decision vector

argmax

Prediction

* 1. Analysis

The final scores for all images are stored and sorted by the score of “unusual” class. There are 400 testing images and 100 of them are “unusual”.

|  |  |  |
| --- | --- | --- |
| Predicted  Actual | Unusual | Usual |
| Unusual | 0.6700 | 0.3300 |
| Usual | 0.1667 | 0.8333 |

Table 5 Confusion matrix for the image classifier

|  |  |  |
| --- | --- | --- |
| Predicted  Actual | Unusual | Usual |
| Unusual | 0.2327 | 0.7673 |
| Usual | 0.0596 | 0.9404 |

Table 6 Confusion matrix for Baseline

|  |  |  |
| --- | --- | --- |
| Predicted  Actual | Unusual | Usual |
| Unusual | 0.7264 | 0.2736 |
| Usual | 0.3298 | 0.6702 |

Table 7 Confusion matrix for the text classifier

|  |  |  |
| --- | --- | --- |
| Predicted  Actual | Unusual | Usual |
| Unusual | 0.5800 | 0.4200 |
| Usual | 0.1533 | 0.8467 |

Table 8 Confusion matrix for the combined classifier