two models use selective search to find out the region proposals, while selective search is a slow and time-consuming process affecting the performance of the network. Distinct from two previous methods, the Faster R-CNN method introduces a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. Furthermore, the model merges RPN and Fast R-CNN into a single network by sharing their convolutional features so that the network can be trained in an end-to-end way. The overall architecture of Faster R-CNN in shown in Figure 5.

4.2 Table Structure Recognition

We leverage the image-to-text model as the baseline. The image-to-text model has been widely used in image captioning, video description, and many other applications. A typical image-to-text model includes an encoder for the image input and a decoder for the text output. In this work, we use the image-to-markup model [Deng *et al.*, 2016] as the baseline to train models on the TableBank dataset. The overall architecture of the image-to-text model is shown in Figure 6.

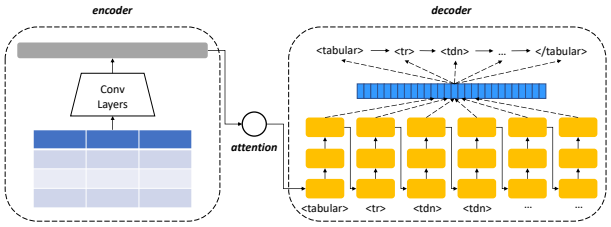

Figure 6: Image-to-Text model for table structure recognition

5 Experiment

5.1 Data and Metrics

The statistics of TableBank is shown in Table 1. To evaluate table detection, we sample 2,000 document images from Word and Latex documents respectively, where 1,000 images for validation and 1,000 images for testing. Each sampled image contains at least one table. Meanwhile, we also evaluate our model on the ICDAR 2013 dataset to verify the effectiveness of TableBank. To evaluate table structure recognition, we sample 500 tables each for validation and testing from Word documents and Latex documents respectively. The entire training and testing data will be made available to the public soon. For table detection, we calculate the precision, recall and F1 in the same way as in [Gilani *et al.*, 2017], where the metrics for all documents are computed by summing up the area of overlap, prediction and ground truth. For table structure recognition, we use the 4-gram BLEU score as the evaluation metric with a single reference.

Task	Word	Latex	Word+Latex
Table detection	163,417	253,817	417,234
Table structure recognition	56,866	88,597	145,463

Table 1: Statistics of TableBank

5.2 Settings

For table detection, we use the open source framework Detectron [Girshick *et al.*, 2018] to train models on the TableBank. Detectron is a high-quality and high-performance codebase for object detection research, which supports many state-of-the-art algorithms. In this task, we use the Faster R-CNN algorithm with the ResNeXt [Xie *et al.*, 2016] as the backbone network architecture, where the parameters are pre-trained on the ImageNet dataset. All baselines are trained using 4×P100 NVIDIA GPUs using data parallel sync SGD with a mini-batch size of 16 images. For other parameters, we use the default values in Detectron. During testing, the confidence threshold of generating bounding boxes is set to 90%.

For table structure recognition, we use the open source framework OpenNMT [Klein *et al.*, 2017] to train the image-to-text model. OpenNMT is mainly designed for neural machine translation, which supports many encoder-decoder frameworks. In this task, we train our model using the image-to-text method in OpenNMT. The model is also trained using 4×P100 NVIDIA GPUs with the learning rate of 0.1 and batch size of 24. In this task, the vocabulary size of the output space is small, including <tabular>, </tabular>, <thead>, </thead>, <tbody>, </tbody>, <tr>, </tr>, <td>, </td>, <cell_y>, <cell_n>. For other parameters, we use the default values in OpenNMT.

5.3 Results

The evaluation results of table detection models are shown in Table 2. We observe that models perform well on the same domain. For instance, the ResNeXt-152 model trained with Word documents achieves an F1 score of 0.9166 on the Word dataset, which is much higher than the F1 score (0.8094) on Latex documents. Meanwhile, the ResNeXt-152 model trained with Latex documents achieves an F1 score of 0.9810 on the Latex dataset, which is also much higher than testing on the Word documents (0.8863). This indicates that the tables from different types of documents have different visual appearance. Therefore, we cannot simply rely on transfer learning techniques to obtain good table detection models with small scale training data. When combining training data with Word and Latex documents, the accuracy of larger models is comparable to models trained on the same domain, while it performs better on the Word+Latex dataset. This verifies that model trained with larger data generalizes better on different domains, which illustrates the importance of creating larger benchmark dataset.

In addition, we also evaluate our models on the ICDAR 2013 table competition dataset. Among all the models, the Latex ResNeXt-152 model achieves the best F1 score of 0.9625, which is better than the ResNeXt-152 model trained on Word+Latex dataset (0.9328). This shows that the domain of the ICDAR 2013 dataset is more similar to the Latex documents. Furthermore, we evaluate the model trained with the DeepFigures dataset [Siegel *et al.*, 2018] that contains more than one million training instances, which achieves an F1 score of 0.8918. This also indicates that more training data does not always lead to better results and might introduce some noise. Therefore, we not only need large scale training data but also high quality data.