Multi-label Image Classification with Visual Attention and Handcrafted Features

Siyi Tang[1], Mingkun Chen[1], Ruge Zhao[2]

[1] *Department of Electrical Engineering, Stanford University*
[2] *Department of Statistics, Stanford University*

Problem & Motivations

- **"iMet Collection" Kaggle challenge**: Predict fine-grained attribute labels for images of museum objects [1]
- **Previous works**
  - CNN models with binary cross-entropy loss assume independence among labels [2]
  - RNN models require pre-defined ordering of labels [3]
- **Our approach**
  - HOG features
  - Visual attention
  - Order-free RNN

Data

- **Official training set**: 109,237 images & 1,103 labels
- Labels belonging to two general categories: *culture* & *tag*

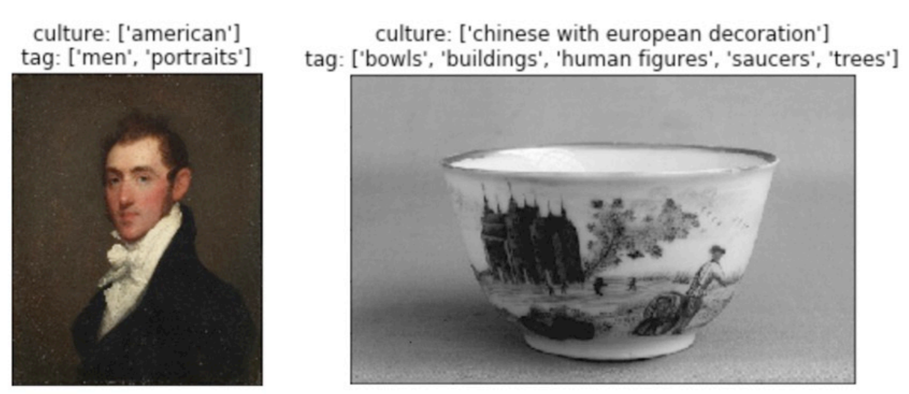

Figure 2: Examples of artwork image with *culture* and *tag* labels

Data Preprocessing

- Discarded labels < 3 occurrences in official training set & split into train/val/test sets with ratio 8:1:1
  - 87,360, 10,920 and 10,920 samples in train/val/test
  - 1,077 labels
- **6x increase** on all images by cropping & resizing to 224 x 224 pixels:
  - Train set images with both *culture* and *tag* labels: 1 resize, 5 random crop
  - Train set images with *tag* labels only: 6 resize
  - Val/test set images: 1 resize, 5 random crop
- **Data augmentation**:
  - Random horizontal flip and random color jitter for train set images
  - Normalized with mean and std of train set

Figure 3: Examples of preprocessed images

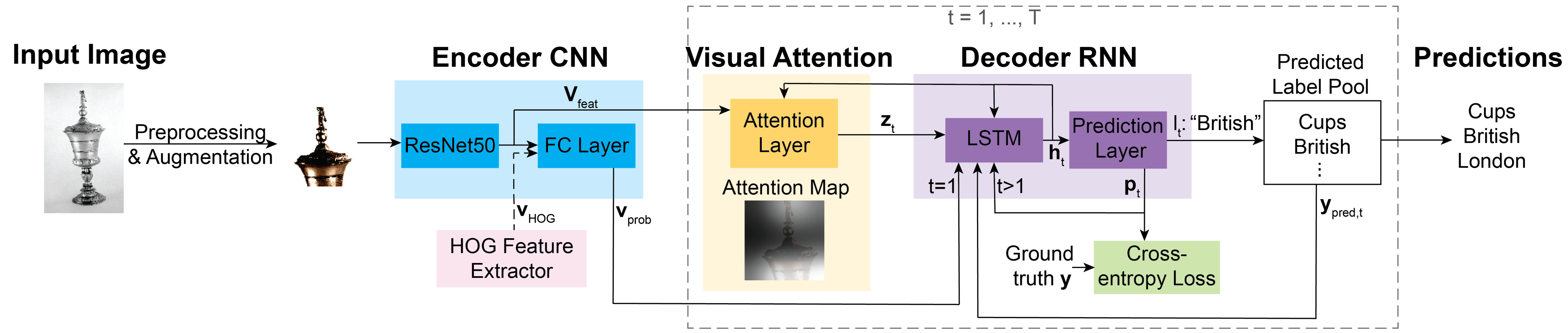

Figure 1. Overview of AttnCNN-RNN (+HOG)

Approach

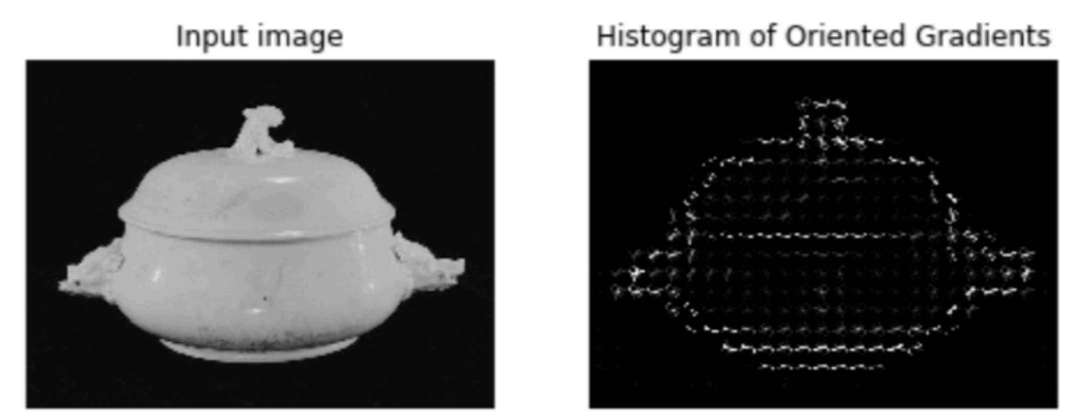

Figure 4: Example of an image and its HOG [5] descriptor

Baseline (ResNet50)

- ResNet50 [4] pretrained on ImageNet
- Modified last FC layer with output size 1,077
- Loss function: cross-entropy, assuming label independence
- Training: froze all previous layers, only trained last FC layer

ResNet50 + HOG

- Same architecture, loss & training procedures as baseline
- HOG features (Figure 4)
- Concatenate ResNet50 last average pooling layer's output with HOG feature vector v_{HOG} before last FC layer

CNN-RNN with Visual Attention (AttnCNN-RNN)

- Encoder CNN - Attention Layer - Decoder RNN [3, 6] (Figure 1)
- **Order-free RNN**:
  - Loss function: $L = -\sum_{j=1}^{1077} p_{t,j} y_j \log(p_{t,j}) + (1-y_j)\log(1-p_{t,j})$
- Terminate prediction when path probability P_t < threshold
  - Path probability: $P_t = \sqrt[t]{Pr(l_1|I) \times \ldots \times Pr(l_t|l_1, \ldots, l_{t-1})}$
  - Class-specific threshold for each class
- **Beam search** at test time

AttnCNN-RNN + HOG

- HOG features + AttnCNN-RNN architecture

Results

Model	F2	F1	Precision	Recall
Baseline (val)	0.327	0.233	0.186	0.587
ResNet50 + HOG (val)	0.337	0.247	0.204	0.563
AttnCNN-RNN (val)	**0.391**	0.263	0.174	0.619
AttnCNN-RNN + HOG (val)	0.390	0.262	0.174	0.617

Table 1: Model Performance on validation set

Model	F2	F1	Precision	Recall
AttnCNN-RNN (test)	0.393	0.264	0.175	0.622

Table 2: Best Model Performance on test set

Analysis

Individual label predictions from ResNet50+HOG

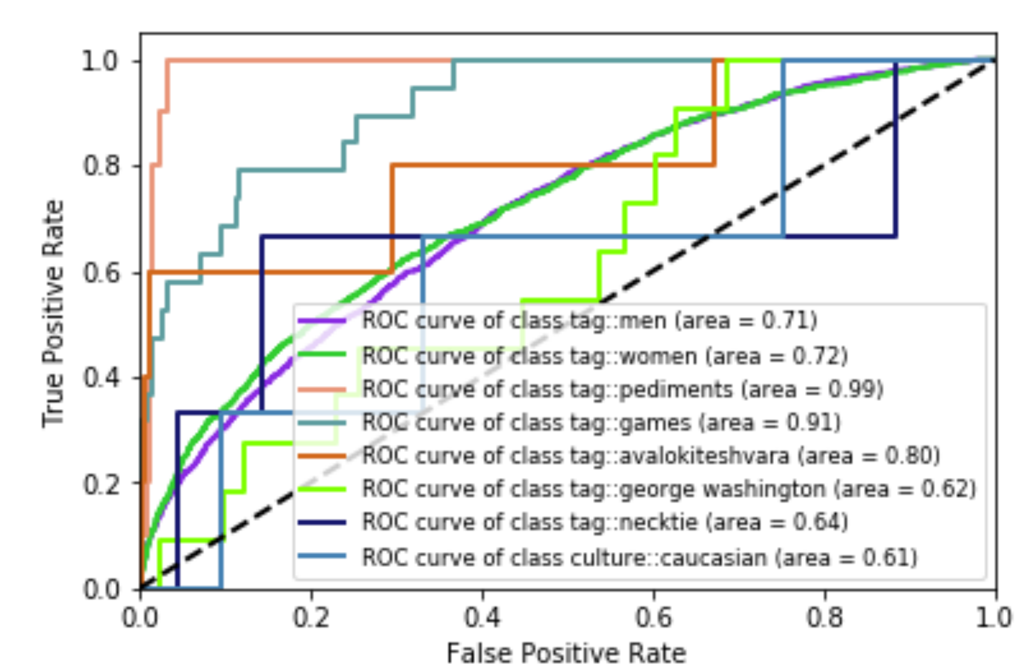

Figure 5: AUC-ROC of ResNet50+HOG on validation set

Attention Visualization for AttnCNN-RNN

Figure 6. Examples of visually attended regions

Prediction Analysis for AttnCNN-RNN

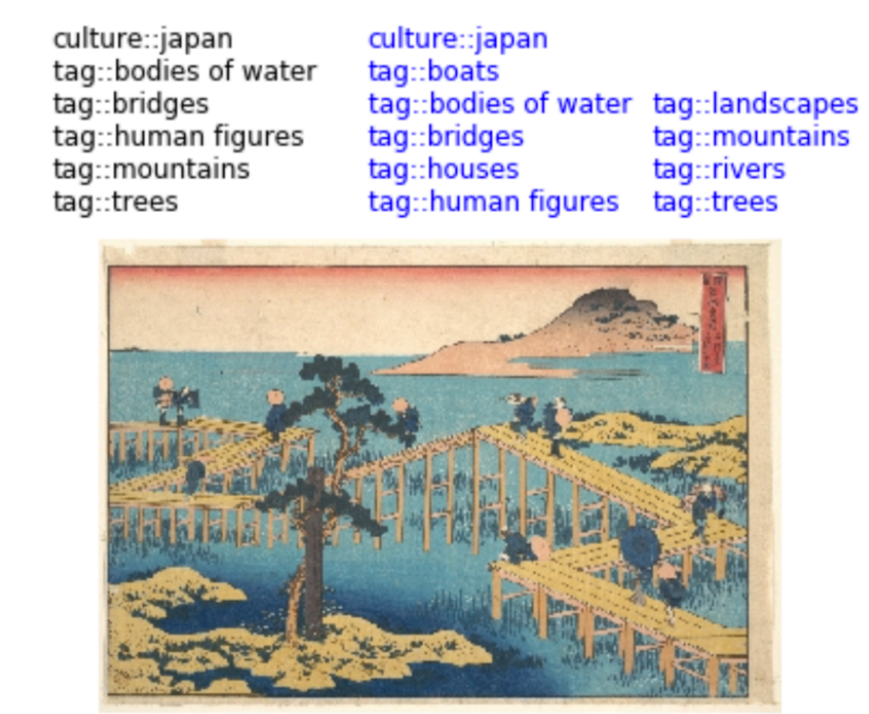

Figure 7. Example of predictions

Conclusion

- Incorporating handcrafted features into CNN provides richer information
- Visual attention mechanism allows our model to focus on image regions associated with the labels
- Use of RNN enables the model to learn inter-dependencies among labels

References

[1] https://www.kaggle.com/c/imet-2019-fgvc6/overview
[2] M.-L. Zhang, et. al., *IEEE Trans Knowl Data Eng*, **18:** 1338 (2006).
[3] K. Xu , et. al., *International Conference on Machine Learning*, **37:** 2048 (2015).
[4] K. He , et. al., *CVPR*, 770 (2016).
[5] N. Dalal , et. al., *CVPR'05*, **1:** 886 (2005).
[6] S.-F. Chen, et. al., *AAAI-18*, (2018).