10-2020

Visual sentiment analysis for review images with item-oriented and user-oriented CNN: Reproducibility companion paper

Quoc Tuan TRUONG
Singapore Management University, qttruong@smu.edu.sg

Hady W. LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Martin AUMULLER
IT University of Copenhagen

Naoko NITTA
Osaka University

Citation

TRUONG, Quoc Tuan; LAUW, Hady W.; AUMULLER, Martin; and NITTA, Naoko. Visual sentiment analysis for review images with item-oriented and user-oriented CNN: Reproducibility companion paper. (2020). *MM '20: Proceedings of the 28th ACM International Conference on Multimedia, Seattle, October 12-16.* 4444-4447.

Available at: https://ink.library.smu.edu.sg/sis_research/5955

Reproducibility Companion Paper: Visual Sentiment Analysis for Review Images with Item-Oriented and User-Oriented CNN

Quoc-Tuan Truong
Singapore Management University, Singapore
qttruong.2017@smu.edu.sg

Hady W. Lauw
Singapore Management University, Singapore
hadywlauw@smu.edu.sg

Martin Aumüller
IT University of Copenhagen, Denmark
maau@itu.dk

Naoko Nitta
Osaka University, Japan
naoko@comm.eng.osaka-u.ac.jp

ABSTRACT

We revisit our contributions on visual sentiment analysis for online review images published at ACM Multimedia 2017, where we develop item-oriented and user-oriented convolutional neural networks that better capture the interaction of image features with specific expressions of users or items. In this work, we outline the experimental claims as well as describe the procedures to reproduce the results therein. In addition, we provide artifacts [1] including data sets and code to replicate the experiments.

KEYWORDS

visual sentiment analysis; convolutional neural networks

ACM Reference Format:
Quoc-Tuan Truong, Hady W. Lauw, Martin Aumüller, and Naoko Nitta. 2020. Reproducibility Companion Paper: Visual Sentiment Analysis for Review Images with Item-Oriented and User-Oriented CNN. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3394171.3414813

1 MODEL

We recall the models proposed in [3]. First, the **base model (VS-CNN)** for visual sentiment analysis is inspired by AlexNet [1]. Figure 1 illustrates the VS-CNN architecture. Input images are preprocessed and cropped to 227 x 227 pixels. It has five convolutional layers (*conv1* to *conv5*) and three fully-connected layers (*fc6* to *fc8*). Unlike AlexNet, the last fully-connected layer *fc8* has only 2 neurons for positive and negative classes, instead of 1000.

Item-Oriented Model (*i*VS-CNN). Hypothesizing that item-specific factors would help identify image sentiment, we allow an item to have its own specific parameters, while sharing most of the parameters with other items. Figure 2 shows two ways to incorporate these into the CNN. The first is to introduce item factors into one of the *convolutional layers*, by dedicating k among n filters to encode the item factors. The modification ensures that those filters can still be in touch with all features from the previous layer and can be used to learn features of the following layer. The second is to introduce the item-orientation in one of the *fully-connected layers*. To represent the highest level of abstraction, we illustrate the modification of the penultimate layer *fc7*. We make k neurons item-specific, while the other $m-k$ neurons remain globally shared.

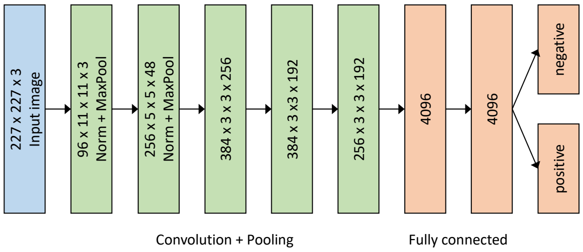
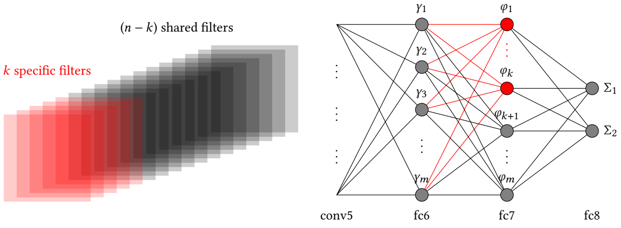


Figure 1: Overall Architecture of VS-CNN



(a) Convolutional Layer **(b) Fully-Connected Layer**

Figure 2: Introducing factors into Conv and FC layers

User-Oriented Model (*u*VS-CNN). Just as we could model item-orientation into the CNN for visual sentiment analysis, we could also model user-orientation into the CNN in an analogous way. To some extent, we seek to capture expressions of sentiments that may be subjective or user-dependent.

2 EXPERIMENTS

We now describe the procedures for conducting experiments to validate three hypotheses as earlier outlined in the original paper.

Dataset. The dataset of review images crawled from *Yelp.com* covers businesses in 7 US cities: Boston, Chicago, Houston, Los Angeles, New York, San Franscisco, and Seattle. As in [2], sentiment classes are derived from ratings: 1 and 2 are considered negative, 3

[1] https://code.preferred.ai/vs-cnn/tree/reproducibility

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3414813

neutral, while 4 and 5 positive. We concentrate on discriminating between positive and negative sentiments. Two balanced datasets *Business* and *User* are created as described in the original paper. From the provided artifacts, the data could be downloaded by running the *download.sh* bash script inside the folder *vs-cnn*:

```
$ chmod +x download.sh | ./download.sh
```

Metrics. We employ three metrics. The first is **Pointwise Accuracy (PoW)**, which tests the ability of a model to label the sentiment for each individual image. The second is **Pairwise Accuracy (PaW)**, which tests the ability of a model to assign a higher probability for a true positive than for a true negative. The third is **Mean Absolute Error (MAE)**, which returns the average difference between the actual label value and the predicted probability.

Environment Setup. For running the experiments, we set up the environment consisting of hardware specification of CPU Intel(R) Xeon(R) E5-2650 v4 @ 2.20 GHz, DRAM 256 GB, GPU NVIDIA Tesla P100. The software requires Python 3.6, CUDA 10.0, and CuDNN 7.5, running on CentOS 7.0. Python dependencies are listed in *requirements.txt* file included in the artifacts and can be easily installed:

```
$ pip3 install -r requirements.txt
```

2.1 Validating Hypothesis 1

The first hypothesis is that sentiment can be detected from images at an accuracy better than random. This can be validated by comparing the base model *VS-CNN* to a random predictor. We train VS-CNN on both *Business* and *User* datasets and evaluate the model across 7 city datasets. This can be done by running *train_base.py* script:

```
$ python3 train_base.py --dataset [user,business]
```

where, *user* and *business* are the corresponding datasets.

For completeness, pre-trained image features are also provided for *Naive Bayes (NB)* classifier. Training and evaluation could be done by running *train_nb.py* script:

```
$ python3 train_nb.py --dataset [user,business]
```

For these balanced datasets, random predictions would achieve 0.5 for all three metrics. Evidently from the results, both VS-CNN and *Naive-Bayes* are able to detect sentiment signals from the images at accuracies greater than 0.5, validating our first hypothesis.

2.2 Validating Hypothesis 2

We hypothesize that *i*VS-CNN with item factors would achieve better performance than VS-CNN that purely relies on visual signals.

The item-oriented model *i*VS-CNN allows item factors to be integrated into either convolutional layer or fully-connected layer. The script *train_factor.py* can test item factors at different levels of abstraction including *conv1*, *conv3*, *conv5*, and *fc7* layers.

```
$ python3 train_factor.py --dataset business
    --factor_layer [conv1,conv3,conv5,fc7]
```

For each setting of *factor_layer*, we get the corresponding results in Tables 1 to 3. Notably, introducing item factor improves upon the performance of the base model VS-CNN at any level of abstraction. Moving the item factor from *Conv1* layer to the *Conv3*, and then to even higher abstraction level *Conv5*, results in further improvements. While there is not much difference between *Conv3* and *Conv5*, modeling item factors in the fully-connected layer FC, which is the highest abstraction, results in the highest performance. These results provide supporting evidence for two points. First, there are slight variances across items when modeling visual sentiments, and taking those into account results in higher accuracy. Second, the item factor seems to be a high-level concept that is better modeled at higher levels of feature abstraction.

Table 1: Item-oriented – Pointwise Accuracy (higher is better)

City	NB	VS-CNN	*i*VS-CNN			
			LowConv	MidConv	HighConv	FC
Boston	0.526	0.534	0.591	0.610	0.601	**0.621**
Chicago	0.554	0.558	0.597	0.620	0.616	**0.628**
Houston	0.549	0.535	0.585	0.606	0.609	**0.615**
Los Angeles	0.537	0.526	0.581	0.602	0.602	**0.608**
New York	0.526	0.530	0.591	0.609	0.603	**0.612**
San Francisco	0.550	0.544	0.584	0.615	0.605	**0.624**
Seattle	0.542	0.534	0.582	0.605	0.597	**0.610**
Avg.	0.539	0.536	0.587	0.609	0.604	**0.615**

Table 2: Item-oriented – Pairwise Accuracy (higher is better)

City	NB	VS-CNN	*i*VS-CNN			
			LowConv	MidConv	HighConv	FC
Boston	0.528	0.568	0.629	0.656	0.650	**0.677**
Chicago	0.557	0.596	0.639	0.667	0.664	**0.696**
Houston	0.569	0.563	0.621	0.655	0.659	**0.669**
Los Angeles	0.552	0.564	0.617	0.647	0.644	**0.662**
New York	0.540	0.564	0.627	0.654	0.653	**0.662**
San Francisco	0.563	0.584	0.624	0.665	0.660	**0.686**
Seattle	0.552	0.576	0.615	0.647	0.642	**0.679**
Avg.	0.551	0.572	0.624	0.655	0.652	**0.673**

Table 3: Item-oriented – MAE (lower is better)

City	NB	VS-CNN	*i*VS-CNN			
			LowConv	MidConv	HighConv	FC
Boston	0.473	0.492	0.411	0.391	0.400	**0.384**
Chicago	0.446	0.489	0.405	0.382	0.387	**0.375**
Houston	0.452	0.491	0.416	0.397	0.394	**0.386**
Los Angeles	0.464	0.492	0.420	0.400	0.400	**0.394**
New York	0.474	0.493	0.412	0.393	0.398	**0.391**
San Francisco	0.450	0.490	0.417	0.387	0.395	**0.379**
Seattle	0.458	0.492	0.420	0.397	0.404	**0.393**
Avg.	0.461	0.492	0.415	0.393	0.397	**0.388**

Table 4: Item-oriented – Comparison between values of k

		*i*VS-CNN			
Metric		LowConv	MidConv	HighConv	FC
Pointwise Accuracy	k = 8	0.575	0.591	0.593	0.611
(higher is better)	k = 16	0.587	0.609	0.604	0.615
Pairwise Accuracy	k = 8	0.606	0.627	0.636	0.671
(higher is better)	k = 16	0.624	0.655	0.652	0.673
MAE	k = 8	0.425	0.410	0.407	0.391
(lower is better)	k = 16	0.415	0.393	0.397	0.388

Table 5: User-oriented – Pointwise Accuracy (higher is better)

City	NB	VS-CNN	*u*VS-CNN			
			LowConv	MidConv	HighConv	FC
Boston	0.537	0.529	0.621	0.622	0.638	**0.654**
Chicago	0.535	0.527	0.637	0.635	0.633	**0.651**
Houston	0.536	0.525	0.610	0.601	0.602	**0.635**
Los Angeles	0.550	0.538	0.634	0.636	0.629	**0.656**
New York	0.541	0.535	0.636	0.626	0.637	**0.661**
San Francisco	0.568	0.548	0.628	0.634	0.633	**0.658**
Seattle	0.528	0.525	0.633	0.605	0.613	**0.643**
Avg.	0.544	0.534	0.631	0.627	0.629	**0.654**

Table 6: User-oriented – Pairwise Accuracy (higher is better)

City	NB	VS-CNN	*u*VS-CNN			
			LowConv	MidConv	HighConv	FC
Boston	0.542	0.539	0.669	0.668	0.694	**0.719**
Chicago	0.554	0.568	0.695	0.696	0.697	**0.739**
Houston	0.569	0.535	0.654	0.641	0.650	**0.703**
Los Angeles	0.561	0.581	0.689	0.690	0.690	**0.735**
New York	0.556	0.562	0.689	0.677	0.701	**0.751**
San Francisco	0.604	0.596	0.679	0.688	0.690	**0.750**
Seattle	0.542	0.541	0.684	0.656	0.681	**0.733**
Avg.	0.562	0.567	0.684	0.679	0.689	**0.737**

Table 7: User-oriented – MAE (lower is better)

City	NB	VS-CNN	*u*VS-CNN			
			LowConv	MidConv	HighConv	FC
Boston	0.464	0.496	0.380	0.381	0.366	**0.353**
Chicago	0.463	0.495	0.365	0.367	0.371	**0.353**
Houston	0.465	0.496	0.391	0.400	0.399	**0.368**
Los Angeles	0.451	0.494	0.368	0.367	0.373	**0.350**
New York	0.460	0.495	0.367	0.377	0.367	**0.343**
San Francisco	0.433	0.492	0.373	0.368	0.370	**0.347**
Seattle	0.471	0.496	0.369	0.395	0.388	**0.360**
Avg.	0.456	0.495	0.371	0.375	0.374	**0.351**

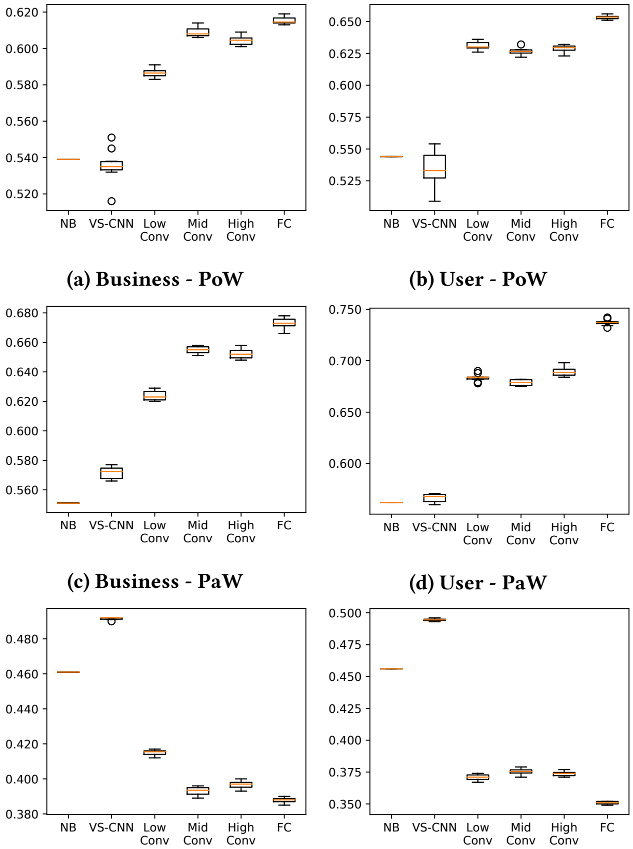
Table 8: User-oriented – Comparison between values of k

		*u*VS-CNN			
Metric		LowConv	MidConv	HighConv	FC
Pointwise Accuracy	k = 8	0.607	0.617	0.613	0.645
(higher is better)	k = 16	0.631	0.627	0.629	0.654
Pairwise Accuracy	k = 8	0.656	0.663	0.657	0.730
(higher is better)	k = 16	0.684	0.679	0.689	0.737
MAE	k = 8	0.395	0.386	0.390	0.359
(lower is better)	k = 16	0.371	0.375	0.374	0.351

2.3 Validating Hypothesis 3

We hypothesize that user factors would respectively play an important role in detecting image sentiments. We compare the user-oriented model *u*VS-CNN to the base model VS-CNN. To test the effect of user factor, we follow the same procedure as testing item factor. The results can be obtained by re-running *train_factor.py* script on the *User* dataset, whose results are shown in Tables 5 to 7:

```
$ python3 train_factor.py --dataset user
    --factor_layer [conv1,conv3,conv5,fc7]
```



(a) Business - PoW

(b) User - PoW

(c) Business - PaW

(d) User - PaW

(e) Business - MAE

(f) User - MAE

Figure 3: Result variation of the experiments.

The performances of the user factor mode *u*VS-CNN on these tables are consistently better than that of the base model VS-CNN. This validates our third hypothesis about the role of modelling user factor in visual sentiment analysis.

2.4 Result Consistency

Originally, the proposed models were implemented based on the Caffe [2] framework, which is now outdated and no longer maintained. For faster training and evaluation, we thus reimplement the models on TensorFlow [3]. Due to some differences between optimization algorithms implemented within the two frameworks, the reproduced results from the current implementation varies slightly from those in the main paper. However, all the hypotheses are still valid and consistent with what have been reported previously.

Figure 3 describes statistics of results from 10 independent runs for each of the experiments. Overall, the variances are relatively low for all of the models, the largest of which is observed for VS-CNN on the *User* dataset. Compared to *Naive Bayes*, VS-CNN captures the positive/negative ranking more effectively as measured by *Pairwise Accuracy*. Both *i*VS-CNN and *u*VS-CNN are stable with extremely low variances. Realizing user/item factors in the most abstract layer *FC7* produce the most effective performance.

[2] https://caffe.berkeleyvision.org
[3] https://www.tensorflow.org

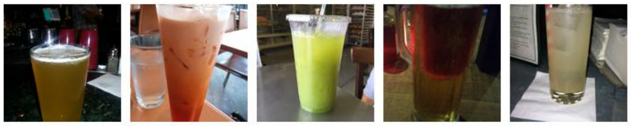
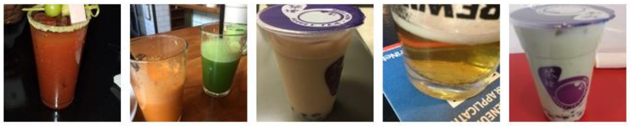


Figure 4: Positive images of drinks, glasses.



Locanda Verde · République · Bubbly Tea · Blue Frog's Local 22 · Bubbly Tea

Figure 5: Images from "contrarian" items reverse the positive images of drinks, glasses.

2.5 Reproducing Tables and Figures

After training and evaluation of models, the results will be written into separate files. We provide *gen_table.py* script to aggregate the results and construct Tables 1 to 8 as reported for ease of comparison. It requires both *i*VS-CNN and *u*VS-CNN trained with $k = 8$ and $k = 16$, and all the output files to be placed in the same folder (i.e., *result_dir*). The tables will be saved in the same folder.

```
$ python3 gen_tables.py --dir result_dir
```

The main paper also shows case studies with illustrative examples (Figures 5 to 10). They were not generated in a completely automatic manner. Here we describe the process in detail and further provide scripts to automate certain parts (where possible).

Figures 5 and 7 show a sample of the most positive and negative images respectively based on the output probabilities of VS-CNN. To retrieve the top 300 images for the positive and negative classes:

```
$ python3 case_study_base.py --num_images 300
    --dataset business
```

The *most_positive* and *most_negative* images will be stored inside *case_study/business* folder [4]. Each training instance may result in slightly different models, potentially resulting in different images.

Each image cluster in Figures 6 and 8 is from "contrarian" items that reverse the sentiment of those images using *i*VS-CNN model. As a reference, we need to provide a cluster of images of similar content, identified manually and stored in the *selected_positive* folder.

As an example, Figure 4 shows 5 images, which have been manually selected from the *most_positive* retrieved above, depicting drinks, glasses, and cylindrical objects. The following command identifies the top N (e.g., $N = 100$) items that would reverse the sentiment of those selected reference images, then retrieves the K (e.g., $K = 5$) most semantically relevant among the negative images in their training data, and saves them in *reverse_positive* folder (as shown in Figure 5):

```
$ python3 case_study_factor.py --num_items 100
    --num_images 5 --dataset business --input_dir
    selected_positive --output_dir reverse_positive
```

As the corresponding output to Figure 4, Figure 5 illustrates the top 5 most similar images to the input, as retrieved from the top 100 "contrarian" items. Specifically, these are the ones with the highest sum of cosine similarities to all the input images. The feature vectors for ranking are extracted from the *fc7* layer of the base model VS-CNN. Similarly, we re-run the script for each of the other manually-clustered set of images and analyze the retrieved photos. Figure 10 in the original paper is to be created by the same procedure but for *User* dataset using *u*VS-CNN model.

We note that the images in Figure 5 differ from the first row of Figure 6 in the original paper, though they share the same concept of drinks, glasses, cylindrical objects. This is due to the automatic selection process that we add for reproducibility as opposed to the originally manual selection. In addition, the model parameters are slightly different after each training, which could lead to changes in the ranking outcome. For reference purposes, we include the images from Figure 4 and 5 as part of the released artifacts.

3 REVIEWING PROCESS

Th present paper is the result of nine rounds (and several months) of discussion between the two reviewers and the authors of the original paper [3]. The initial reviews pointed out some inconsistencies in the reproduced results, which were addressed by the authors (cf. Section 2.4).

After these inconsistencies had been fixed, the automated reproduction of "contrarian items", as presented in Figure 5, was a key obstacle in terms of reproducibility of the original paper. The authors came up with several different suggestions which eventually allowed for an automated selection that—albeit different to the original results—allowed all figures and tables of the original paper to be reproduced by running the scripts mentioned in this paper.

The present state of the artifacts allowed the reviewers to reproduce all tables and figures found in [3], and differences to the original results are well-discussed in this paper.

4 CONCLUSION

We investigate the roles of item-orientation and user-orientation for CNN-based visual sentiment analysis. The experiments show that the item factor model *i*VS-CNN achieves higher accuracy than the base VS-CNN, particularly when item factor is incorporated at higher levels of abstraction. Experiments for user factor, *u*VS-CNN model, also yield similar trends supporting our hypothesis about the role of users. The experimental results can be replicated with the provided artifacts, which reinforces the validation of our claims.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its NRF Fellowship Programme (Award No. NRF-NRFF2016-07).

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*. 1106–1114.

[2] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP*. 79–86.

[3] Quoc-Tuan Truong and Hady W Lauw. 2017. Visual sentiment analysis for review images with item-oriented and user-oriented CNN. In *ACMMM*. 1274–1282.

[4] Images in Figure 9 could be sampled by running the same script for *User* dataset.