Accepted to be Published in: Proceedings of the 2019 IEEE International Conference of Image Processing (ICIP), September 22-25, 2019, Taipei, Taiwan.

JOINT OBJECT AND STATE RECOGNITION USING LANGUAGE KNOWLEDGE

Ahmad Babaeian Jelodar, and Yu Sun

Department of Computer Science and Engineering, University of South Florida, Tampa, FL, USA

ABSTRACT

The state of an object is an important piece of knowledge in robotics applications. States and objects are intertwined together, meaning that object information can help recognize the state of an image and vice versa. This paper addresses the state identification problem in cooking related images and uses state and object predictions together to improve the classification accuracy of objects and their states from a single image. The pipeline presented in this paper includes a CNN with a double classification layer and the Concept-Net language knowledge graph on top. The language knowledge creates a semantic likelihood between objects and states. The resulting object and state confidences from the deep architecture are used together with object and state relatedness estimates from a language knowledge graph to produce marginal probabilities for objects and states. The marginal probabilities and confidences of objects (or states) are fused together to improve the final object (or state) classification results. Experiments on a dataset of cooking objects show that using a language knowledge graph on top of a deep neural network effectively enhances object and state classification.

Index Terms— State Classification, Transfer Learning, joint object and state classification, Concept-Net.

1. INTRODUCTION

Image classification is a research area in computer vision that has gained great attention in recent years mainly to tackle object classification and detection problems [1, 2, 3]. Object states, on the contrary, have not been considered as much as object classification in recent literature. Moreover, object states require further analysis especially for robotics-based applications. Robotic manipulation, task planning, and grasping require knowledge and constant feedback about the state of the environment and objects. For instance, if a robot chef wants to perform the task of chopping an onion, it has to grasp the whole onion, cut it into half, recognize its new state (sliced), grasp it accordingly, and cut it into smaller parts while continuously monitoring the state. Ultimately, the robot needs to recognize the desired state and understand when it has reached the end of the procedure (e.g. chopping). The problem of states has been analyzed in several previous works [4, 5, 6]. Similar to [6] we will address the issue of states in cooking related images.

States of objects are not independent of the object itself, the action happening, or the scene. Additional information from a single image such as knowledge about the objects in the image will lead to more accurate state classification results. Some research has focused on joint state and action or state and object classification [2]. Language knowledge graphs are useful for analyzing semantic relationships [7, 8]. Language knowledge graphs can draw a connection between objects and states in an image and define the likelihood of an object and state occurring together. Combining the image classification power of deep convolutional networks with the semantic power of a language knowledge can provide a powerful tool for joint state and object classification.

In this paper, we present a pipeline consisting of a deep convolutional network and a language knowledge graph inference strategy for joint state and object classification as shown in Figure 1. The Resnet-50 architecture from [1] is trained with two parallel classification layers for object (e.g. potato) and state (e.g. diced) classification. Joint confidences for each pair of object and states are computed using the relatedness assertion in Concept-Net. The object and state marginal probabilities are computed using the confidences from the deep network and the joint confidences derived from Concept-Net. The outputs from Resnet-50 are concatenated with the inferred marginal probabilities which are then fed to two multilayer perceptron (MLP) networks. The MLPs are trained and the whole pipeline is evaluated over a dataset of cooking objects. A selector gate is trained to predict whether the CNN model will predict correctly or incorrectly given an input image and is incorporated in the model for prediction improvement. Our work has two main contributions:

- A new pipeline for joint state and object classification which incorporates language knowledge to help with state and object predictions.
- A selector gate that improves classification accuracy by utilizing the input and output of a trained classifier.

The rest of the paper is organized as follows. Section 2 introduces the related work in state classification. Section 3 introduces the methodology including the language knowledge used for state and object classification. Section 4 discusses experiments and results and Section 5 concludes the paper.

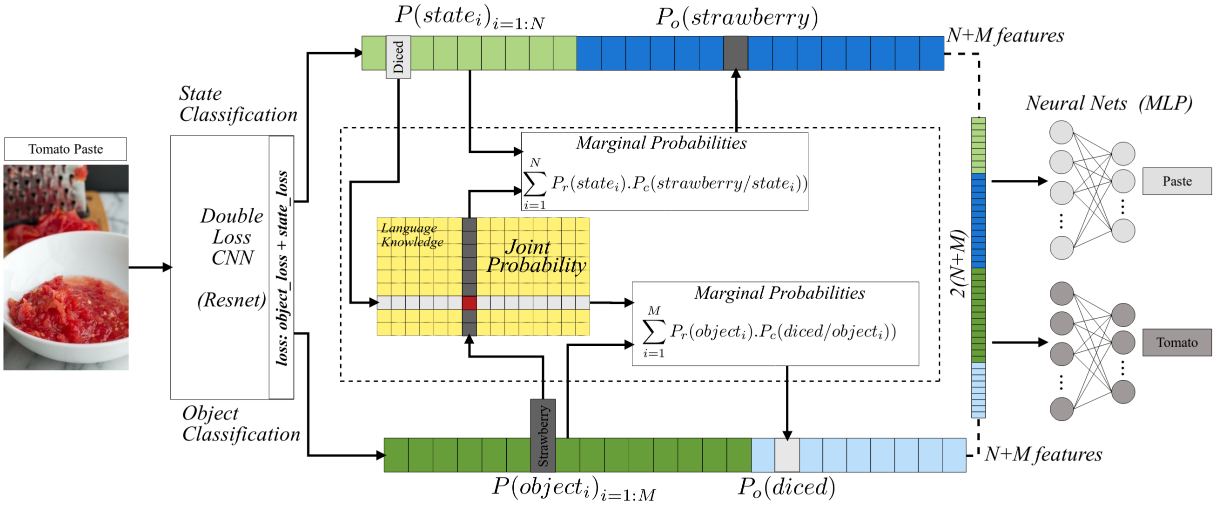

Fig. 1. Pipeline for State and Object Classification using Language Knowledge.

2. RELATED WORK

Object classification and detection are very popular areas of research [1, 9, 10, 11], but state classification from a single image requires more investigation. Some work explicitly address the states [6], and some perform state identification implicitly [12]. In [2], action attributes and parts are used as states of an action for action classification. High level image attributes have also been incorporated in CNNs and LSTMs to provide descriptions for an image [3]. In [6], a dataset of states for cooking objects was introduced and the problem of state classification in cooking related images was addressed. We use this dataset in our work. In [13], states and state transformations between objects including cooking objects are analyzed on a collection of images.

In [4], states of objects and state-modifying actions are jointly detected using a discriminative clustering cost. In [5], a multi-task CNN is proposed for binary attribute prediction. Each binary attribute can be considered as a state in our context. In [14], a deep convolutional and recurrent framework is presented for providing multiple object labels from a single image. This work is similar to our work in the aspect that it provides multiple labels for a single image. Facial expression can also be considered as a state of the face. In [15], a multi-loss architecture is proposed to capture both identity and expression associated features for face expression classification. In [16], a framework is proposed that simultaneously models multiple concepts or states of position in a sequence using an RNN and a spatio-temporal graph.

Knowledge representations have been effectively used in combination with classification approaches. In [17], a video understanding framework was proposed that deploys a deep convolutional network together with a knowledge representation. Knowledge representations can also be incorporated in robotics applications [18] and to aid robots in manipulation decisions for cooking actions [19, 20]. Concept-Net has also been employed for object detection. In [7], semantic consistency is seeked by combining information from a knowledge graph such as Concept-Net and any object detection algorithm.

3. THE PIPELINE

We propose a pipeline for joint state and object identification. The pipeline includes a convolutional neural network, a language model and two MLP networks as shown in Figure 1. We apply a selector gate on the pipeline outputs to improve results as depicted in Figure 2.

3.1. Stage 1: Double Loss Convolutional Network

In the first stage of the pipeline, we use the Resnet architecture with two outputs- one for state and one for object classification. The two applications use the same weights apart from the last layer. The loss applied for object and state classification are defined separately and trained simultaneously. The network outputs two different sets of confidences via the soft-max layer, one for the state classes, $[P(state_i)_{i=1:N_{states}}]$, and another for the object classes, $[P(object_i)_{i=1:N_{objects}}]$, as shown in Figure 1. The notations N_{states} and $N_{objects}$ are the number of states and objects respectively. The soft-max confidences are the first set of probabilities we obtain for object and state classification. We name them as prior probabilities of each object (or state) occurring in the image.

3.2. Stage 2: Language Knowledge based Features

In natural language processing, documents, sentences, and words are processed to extract meanings, relationships and word embeddings. In this paper we will use the ***Concept-Net***, which is more powerful than the widely used Word2vec [8], and the ***Google N-gram Viewer*** to quantify word relations.

Concept-Net is a language knowledge graph that includes words and phrases as nodes and natural language relationships between the nodes as edges [21]. Concept-Net defines and implements a class of language- and source-independent relations between words and phrases including *IsA*, *UsedFor*, and *CapableOf* and also associates weights with every relationship. Weights of relations are calculated based on an aggregation of weights from various sources. We use the weights from the *RelatedTo* relation (or assertion) of the Conceptnet API to quantify the relationship of a specific state (e.g. sliced) with a specific object (e.g. bread).

In natural language processing, an N-gram is a sequence of N items (e.g. words) in a bed of various documents called a corpus [22]. The frequency of two or multiple words happening together (N-grams), can be representative of how related they are. The Google N-gram Viewer is a Google based search engine that shows the frequency of any N words occurring consecutively in Google's text sources [23]. We use the frequencies extracted from the Google N-gram Viewer to represent the relationship between states and objects.

3.2.1. *Feature Extraction*

The correct identification of objects is associated with the correct identification of states and vice versa. We use the Concept-Net and the Google N-gram Viewer to quantify the relationship between the states and objects in the dataset. We first define a set of words associated with each object and a set of words associated with each state. For instance, for the object $potato$ we define the set $\{potato, potatoes\}$ and we define $\{creamy, paste, mashed, mash, softened, whipped\}$ as the set representing the state $creamy$. To calculate the joint probability of an object (e.g. potato) and a state (e.g. creamy), every pair of object and state from the two sets is looked up in Concept-Net or the N-gram Google Viewer to derive a relatedness value. The maximum and the mean values for each pair are recorded (e.g. potato-creamy). The confidences are normalized so that the sum of all probabilities of a state over various objects and the sum of all probabilities of an object with different states each sum up to 1.

We calculate the marginal probabilities for each object assuming the state prior probabilities and joint (conditional) probabilities ($[P(object/state_i)_{i=1:N_{states}}]$) derived from the language knowledge source (e.g. Concept-net or Google N-gram Viewer). We conversely compute the marginal probabilities for the states using joint (conditional) probabilities ($[P(state/object_i)_{i=1:N_{objects}}]$). The relations for marginal probabilities for each object $P(object)$, and state, $P(state)$, is given in (1), and (2) respectively.

$$P_o(object_j) = \sum_{i=1}^{N_{states}} P_r(state_i).P_c(object_j/state_i)) \quad (1)$$

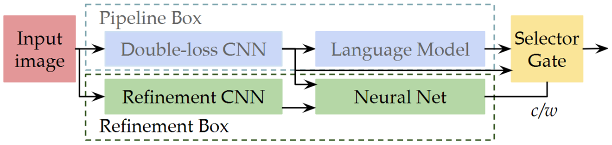

Fig. 2. Pipeline Refinement.

$$P_o(state_j) = \sum_{i=1}^{N_{objects}} P_r(object_i).P_c(state_j/object_i) \quad (2)$$

In (1), and (2), P_c is the conditional probability of an object in respect to a state or vice versa which is derived from the language knowledge, P_r is the output confidence from the Resnet, and P_o is the marginal probability.

3.3. Stage 3: Neural Network Predictions

The marginal and prior probabilities are concatenated together to create a feature vector of size $2 \times N_{objects}$ and $2 \times N_{states}$ for objects and states respectively. The concatenated object and state features are merged together to create a final feature vector with size $V_{final} = 2 \times (N_{states} + N_{objects})$. The feature vector V_{final} is given as input to two separate MLP networks for object and state classification respectively as shown in Figure 1. A three layer MLP is selected using the validation set as the finalized architecture of the networks.

3.4. Stage 4: Model Refinement

The pipeline converts correct predictions into incorrect predictions in some cases. To reduce these conversions, a refinement procedure is proposed that starts training after the double loss CNN has finished training. The refinement model is trained to predict the probability of an image being classified correctly by the pipeline. The refinement model contains a Resnet-based CNN which returns two outputs (classes) for a given input image; one output represents an image being classified correctly and the other represents the image being classified incorrectly. The two outputs are associated with two confidences. The two confidences are concatenated with the ouput probabilities from the double loss CNN from the pipeline. Two separate neural networks are trained for correct/incorrect object (and state) probability predictions using the concatenated feature vectors. The outputs from the MLP are used as a selector for a gate selector block. A value of one for the selector output, means that the initial prediction is correct and the object (or state) confidences from the double loss CNN are used for predictions. A value of zero for the selector output, means that the the probabilities after language knowledge incorporation should be used for predictions. The refinement model is depicted in Figure 2.

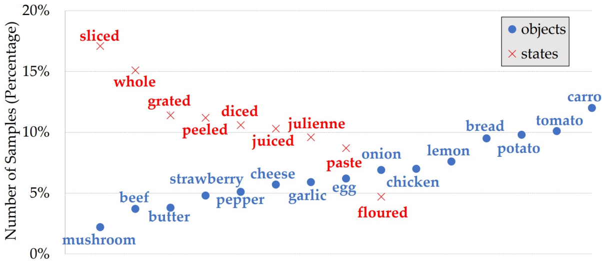

Fig. 3. States and Objects statistics in the dataset.

4. EXPERIMENTS AND RESULTS

4.1. Dataset

We used the state classification dataset from [6]. It consists of images from 15 cooking objects and 11 state classes as shown in Figure 3. For our experiments, we removed the states *mixed* and *other* that are not associated with any specific type of object. When training, we use data augmentation to balance the classes and compute the accuracy as average class accuracy. We annotate the dataset with object labels. The total number of images in the dataset is around 9.5K. – 70% train, 15% validation, and 15% test set. The dataset includes an online challenge page with the best state classification results ranked from best to worst[1]. We used the statistical information from the knowledge representation in [19] to derive the most frequent objects and states represented in cooking events. States were analyzed hierarchically, and the main states associated with the most frequent objects were derived [6].

4.2. Results

We implemented the Resnet model in Tensorflow and initialized with pre-trained weights from Imagenet. The single classifier layer was removed and a double classifier layer was added for states and objects. We trained the model for 15 iterations and with an initial learning rate of 0.01. Only weights from the last block of Resnet were trained and the rest were kept frozen. The relatedness values of objects and states were downloaded from the Concept-Net (or Google N-gram Viewer) Web APIs using the Python Request library and the normalized versions of the relatedness values were recorded as joint probabilities. The final features were then computed and then given to MLPs as mentioned in Subsection 3.3.

We compared the pipeline with other methods and report the results in Table 1. We compared the pipeline with the raw initial confidences, the linear combination of the initial confidence and the marginal probabilities from Concept-Net, and an SVM-based version of the pipeline. The results show that all methods containing a language knowledge outperform the Resnet network as shown in Table 1. The neural network based method that uses features from the Resnet output and the Concept-Net features outperforms all other methods. Results in Table 1 show that self-correction using the refinement model improves the results even further.

Table 1. States and object classification accuracy on the test set with and without using Concept-Net (Concept-Net as CN, Google N-gram as GN).

Model	States	Objects
Resnet	79.4%	74.1%
(Resnet,CN) + SVM	79.7%	74.2%
(Resnet,GN) + MLP	80.1%	74.2%
(Resnet,CN) + MLP	80.4%	74.3%
(Resnet,CN) + MLP + Refinement	**80.9%**	**75%**

Figure 4 shows an instance of an incorrect result (diced strawberry) converting to a correct result (tomato paste) when using Concept-Net. Concept-Net can make mistakes. For example grated butter has a high relatedness confidence in the Concept-Net graph although in the real world it is unlikely to see grated butter often. Therefore, it is easy to flip a correct *creamy butter* to an incorrect *grated butter*. The refinement model has the ability to prevent some of these cases.

	Object CNN	
1	Strawberry	0.56
2	***Tomato***	0.43
3	Beef / Pork	0.01
	...	...
15	Mushroom	0

	Object: CNN+CN	
1	***Tomato***	0.58
2	Strawberry	0.39
3	Beef / Pork	0.01
	...	...
15	Orange	0

	State CNN	
1	Diced	0.52
2	***Paste***	0.47
3	Whole	0.01
	...	...
9	Peeled	0

	State: CNN+CN	
1	***Paste***	0.53
2	Diced	0.45
3	Whole	0.01
	...	...
9	Floured	0

Fig. 4. Objects and states CNN probabilities vs Concept-Net (CN) probabilities. Probability of diced strawberry is lower than tomato paste when CN is used.

5. CONCLUSION

The states of a cooking object are valuable information for a robot chef when performing cooking events and are closely related with the object itself. This paper presented a deep neural network with two joint losses for object and state classification. A language knowledge graph was deployed on top of confidences from a double loss CNN for extracting language based confidences. A MLP-based classifier was trained using the combination of confidences from both stages. Experiments on a state classification dataset consisting of cooking objects showed that using a language knowledge together with the confidences from the deep network improved both object and state classification performance.

[1] http://rpal.cse.usf.edu/datasets_cooking_state_recognition.html

6. REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CVPR*, pp. 770–778, 2016.

[2] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," *ICCV*, pp. 1331–1338, Nov 2011.

[3] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," *ICCV*, vol. 00, pp. 4904–4912, Oct. 2018.

[4] J. B. Alayrac, J. Sivic, I. Laptev, and S. Lacoste-Julien, "Joint discovery of object states and manipulation actions," *ICCV*, 2017.

[5] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia, "Multi-task cnn model for attribute prediction," *IEEE Transactions on Multimedia*, vol. 17, pp. 1949–1959, 2015.

[6] A. B. Jelodar, M. S. Salekin, and Y. Sun, "Identifying object states in cooking-related images," *arXiv preprint arXiv:1805.06956*, May 2018.

[7] Y. Fang, K. Kuan, J. Lin, C. Tan, and V. Chandrasekhar, "Object detection meets knowledge graphs," *IJCAI-17*, pp. 1661–1667, 2017.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *ICLR*, May 2013.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS*, vol. 1, pp. 1097–1105, 2012.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CVPR*, 2015.

[12] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," *CVPR*, 2016.

[13] P. Isola, J. J. Lim, and E. H. Adelson, "Discovering states and transformations in image collections," *CVPR*, 2015.

[14] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," *CVPR*, 2016.

[15] Z. Meng, P. Liu, J. Cai, S. Han, and Y. Tong, "Identity-aware convolutional neural network for facial expression recognition," *FG 2017*, pp. 558–565, May 2017.

[16] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," *CVPR*, pp. 5308–5317, June 2016.

[17] A. B. Jelodar, D. Paulius, and Y. Sun, "Long activity video understanding using functional object-oriented network," *IEEE Transactions on Multimedia*, pp. 1–12, 2018.

[18] D. Paulius and Y. Sun, "A survey of knowledge representation in service robotics," *Robotics and Autonomous Systems*, 2019.

[19] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun, "Functional object-oriented network for manipulation learning.," *IROS*, pp. 2655–2662, 2016.

[20] D. Paulius, A. B. Jelodar, and Y. Sun, "Functional object-oriented network: Construction & expansion," *ICRA*, pp. 1–7, May 2018.

[21] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," *AAAI*, 2016.

[22] C. D. Manning and H. Shutze, "Foundations of statistical natural language processing," in *The MIT Press*, 1999.

[23] Google, "Google ngram viewer," http://books.google.com/ngrams/datasets, 2012.