CNN visualization with Grad-CAM

Ruihao Wang, Liujun Zhang, Yijie Zhou, Yu Guo

Abstract

Convolutional Neural Network (CNN) is one of the popular deep learning methods today. It has been applied in many areas such as image classification, object segmentation, natural language processing and reinforcement learning. Researchers have been studying visualizing the CNN model to characterize its classes and features during training with different algorithms. In this report we refer to one paper "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization" about Gradient-weighted Class Activation Mapping (Grad-CAM). This paper introduces an approach to use the gradients of any target concept flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept. We reproduced this approach by vgg-16 model with a 4-class animal images dataset. During our research process, we discover that the visualized features/classes from Grad-CAM can be reproduced with a higher resolution by removing the pooling layers in the CNN model. Hence, we retrain the model with non-pooling layer structure to prove our idea. To make the demonstration concise, we replace the Fully Connected Layers by Global Average Pooling (GAP) to simplify the model. GAP is an effective approach in CNN model that can reduce the dimension of feature maps. It is also stated in the paper "Learning Deep Features for Discriminative Localization" which introduces the basic idea of CAM.

Part 1: Summary of the original paper

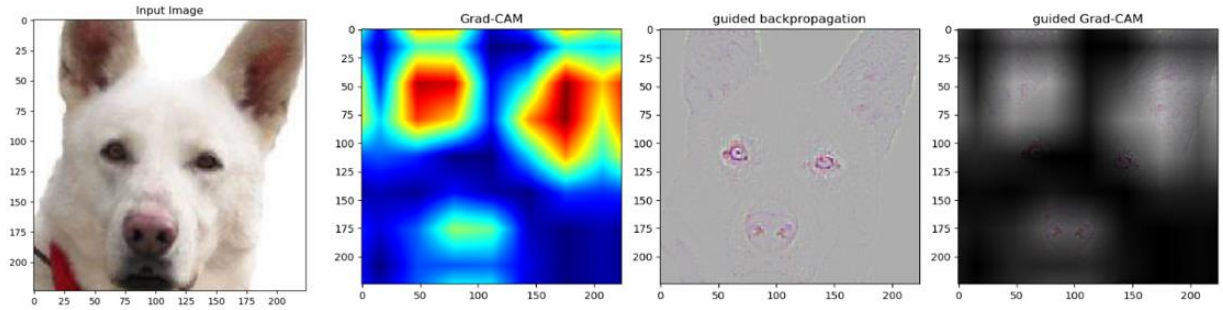
In the paper, the authors provide a mapping approach to highlight the region in the image the prediction relies on. In the CNN model, feature maps are the product of convolutional layers. These feature maps correspond to different features, textures and patterns. The core of Grad-CAM is to use the gradient of the score for the target class, with respect to each feature maps to generate the weight "a", The weight "a" represents a partial linearization of the deep network

downstream from overall feature maps and captures the ‘importance’ of each feature map for the target class. After the weighted combination based on weight “a” and feature maps, the ReLU activation function is used to obtain the heat-map of the desired class. Coordinating with the heat-map, the Guided Backpropagation is used by multiplication to perform the fine-grained importance like pixel-space gradient visualization with a relatively high resolution.

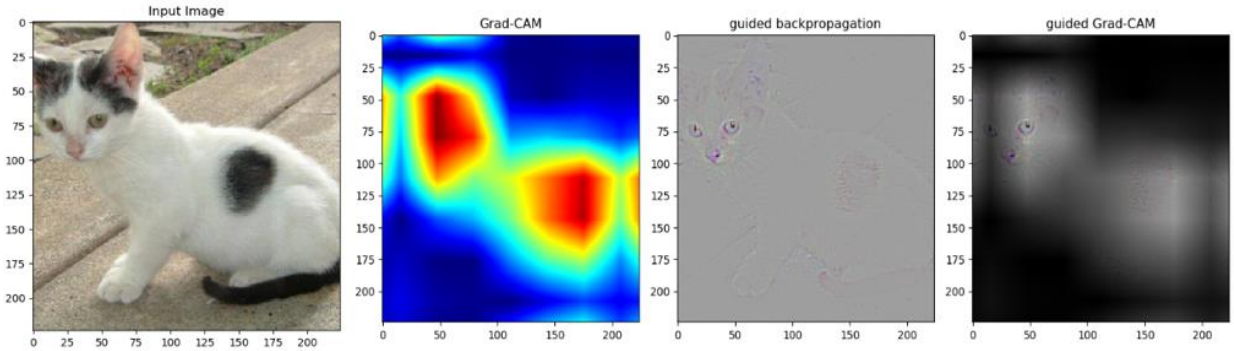
Part 2: Reproduce Result

We search online to build our own dataset. In this dataset, we have 4 classes images: dog, cat, horse and bird. The numbers of images of each class are: 12500, 12500, 8452, 8671. We use vgg-16 to classify the images in our dataset and then use grad-cam approach to generate the heat-maps. We run our model on a desktop with a Nvidia RTX 2080 with the IDE PyCharm. Our code is inspired by this tutorial(https://github.com/insikk/Grad-CAM-tensorflow). The TensorFlow is used as a machine learning library. Here are some examples of the reproduction results:

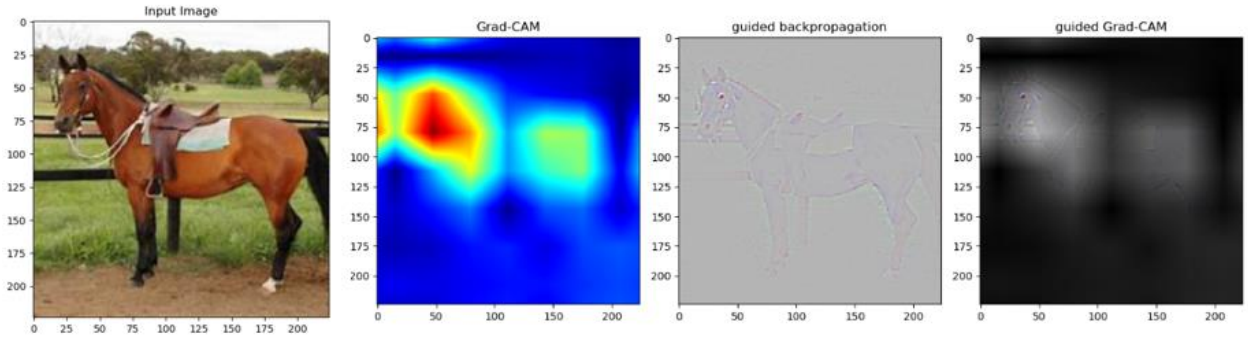
Class: Dog



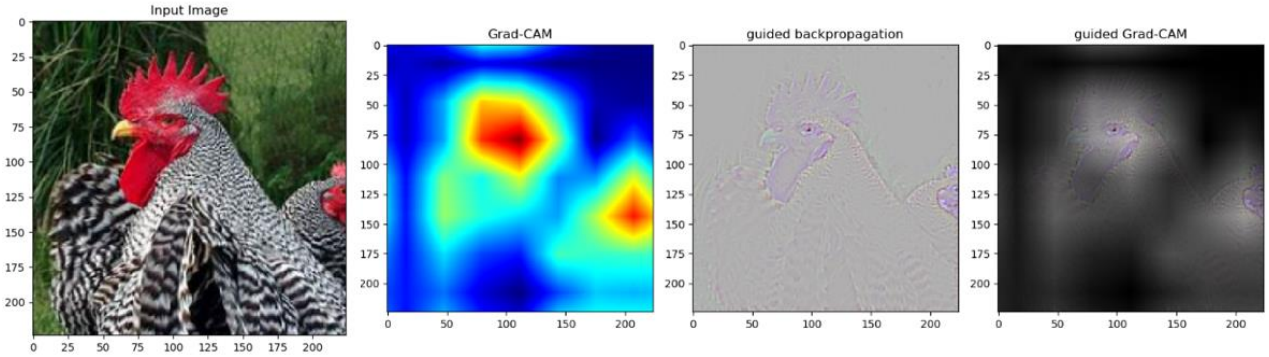
Class: Cat



Class: Horse



Class: Bird

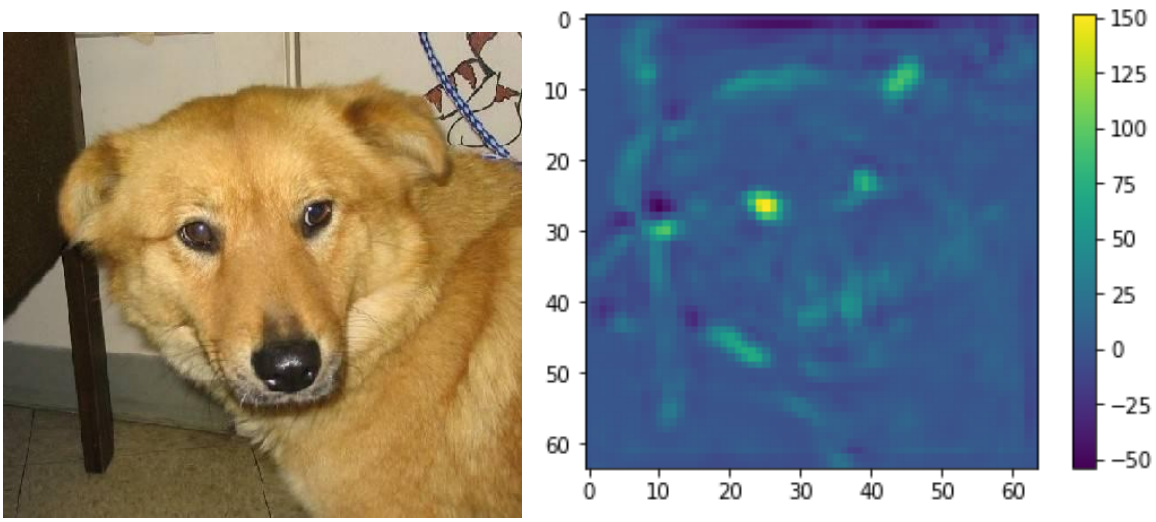


Part 3: Discovery and Demonstration

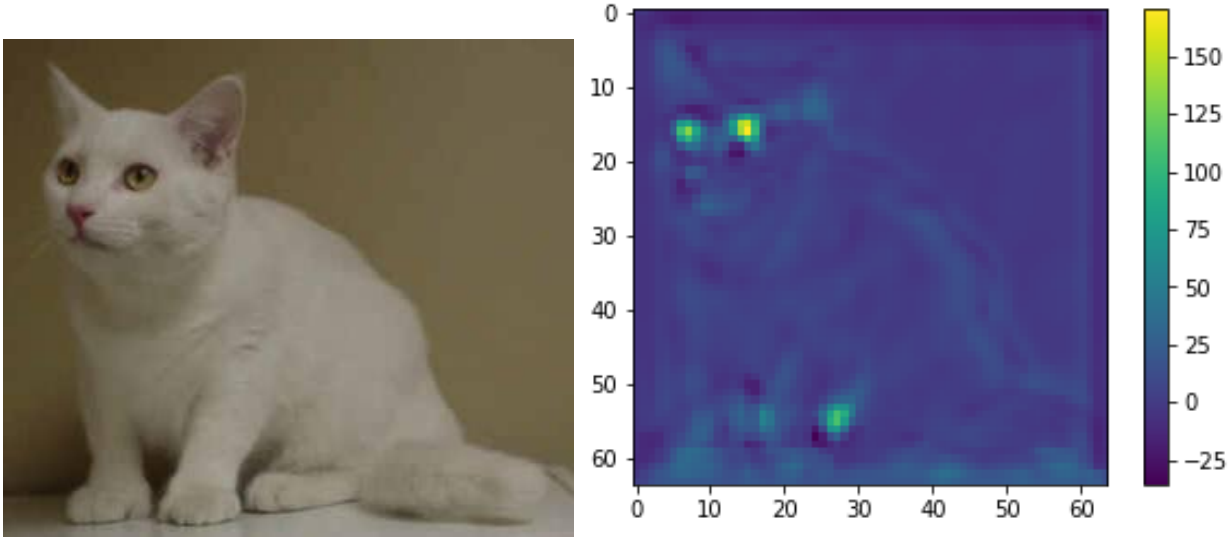
During our work process, we find that Grad-CAM generates the heat-map with relatively low resolution, which means that it only provides a rough region in the image. Although by fusing the Guided Backpropagation and the heat-map, the resolution is improved to show detailed information, we think we can achieve a similar performance by removing the pooling layers inside the CNN model. In this case, the heat-map has the same size as original input. It is not needed to “forcefully” increase the size of heat-map of the original input anymore. We run a demo based on this approach. First, we build the CNN model without any pooling layer after the convolutional layers. Second, we use Global Average Pooling (GAP) to replace the fully connected layers. Third, we train this model with our 4-class dataset and get the final model and weights with 90% accuracy . Fourth, the weights between the GAP and final output are picked up to generate the weighted combination of each feature map. The final heat-map is done by

summing these weighted feature maps. The code is already uploaded to the project website. Here are some examples from the demo:

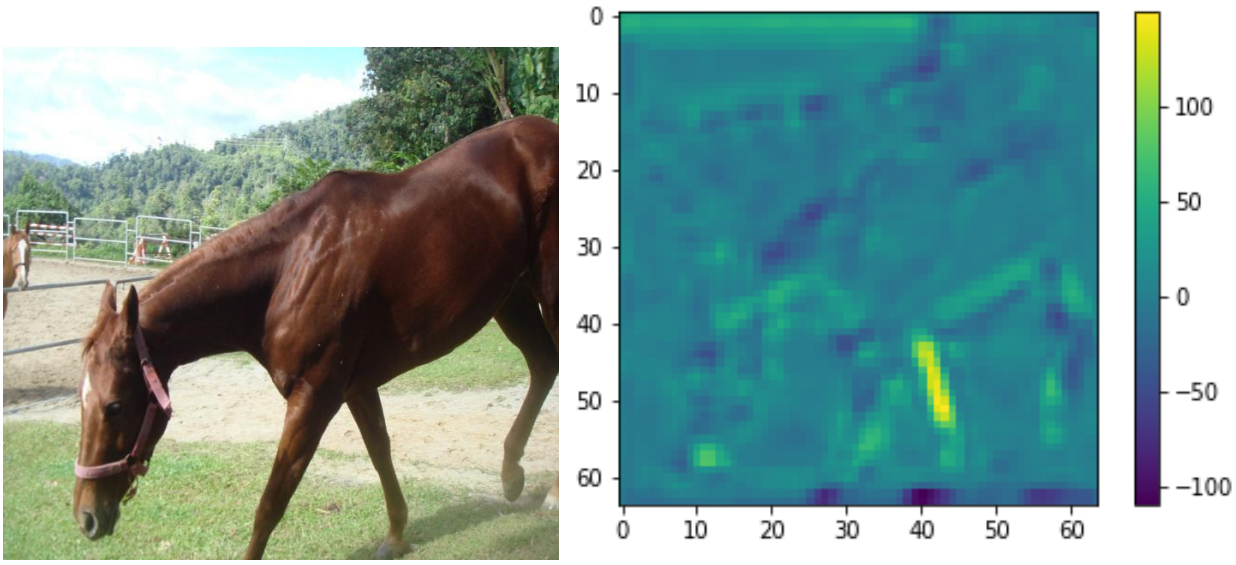
Class: Dog



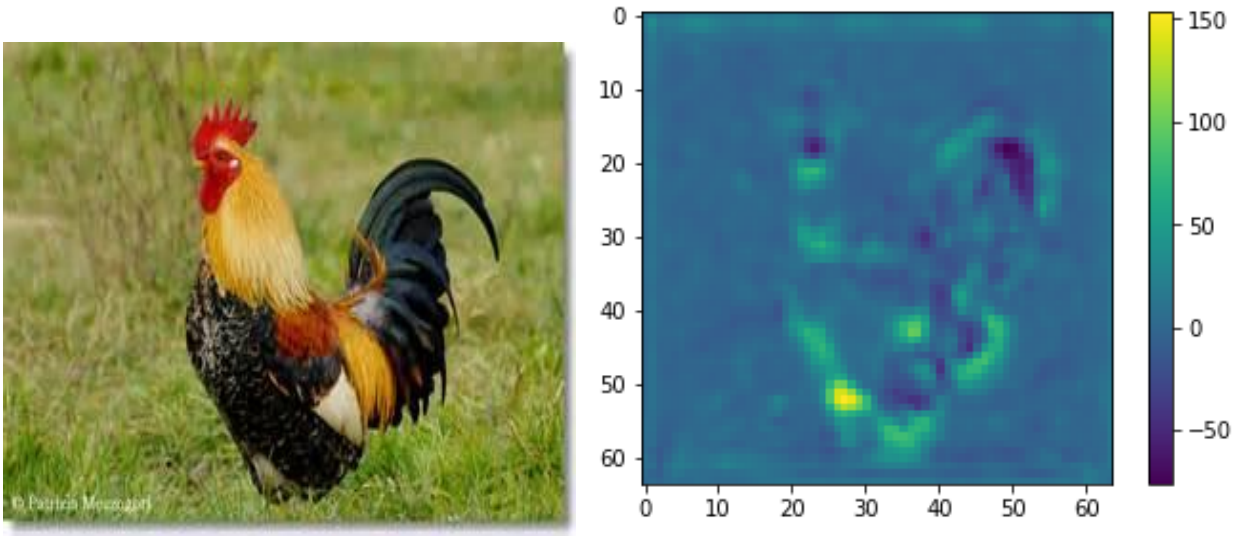
Class: Cat



Class: Horse



Class: Bird



Part 4: Discussion and Reference

The Grad-CAM is an updated version of the normal CAM in CNN visualizing task. In the CAM paper, the author provides an approach that combines the GAP weights with feature maps to show the Area of Interest (AOI) when CNN classifies the image. However, this approach will change the structure of the original CNN model. It is not suitable for some structure-fixed model like VGG. In this case, the Grad-CAM solves this problem perfectly. In Grad-CAM, the model structure remains the same. Based on the mathematical prove, the weights “a” in Grad-CAM is

proportional to the weights in CAM so that we can better understand how the gradients of target class with respect to feature maps will represent the importance of each feature maps for the target class.

Here is a list of relevant papers:

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. CVPR'16 (arXiv:1512.04150, 2015).

M. Lin, Q. Chen, and S. Yan. Network in network. International Conference on Learning Representations, 2014.

Part 5: Conclusion

Based on our reproduced results, we find that it corroborates the claims of the original paper. With a well-trained model, Grad-CAM generates the heat-map for different target class and the heat-map containing independent and specific features of the target class. It is a good approach for visualizing the CNN model.