

AI 기반 영상 데이터 분석 실습

- Day 5 -

Course overview

Course overview

Day	Morning session (이론)	Afternoon Lab session (실습)
Jan. 19 (Mon)	<ul style="list-style-type: none">인공지능 개요Vision task 소개MNIST Classification review	<ul style="list-style-type: none">GitHub 활용법주제 선정데이터 수집
Jan. 20 (Tue)	<ul style="list-style-type: none">영상처리 기초이미지 전처리 기법	<ul style="list-style-type: none">전처리, 증강, 시각화Dataset splitData set class & Data loading
Jan. 21 (Wed)	<ul style="list-style-type: none">Fundamentals on CNNBasic architectures	<ul style="list-style-type: none">CNN architecture 구현 (Resnet)
Jan. 22 (Thu)	<ul style="list-style-type: none">Weight update (Gradient descent, Optimizers)Loss monitoring	<ul style="list-style-type: none">Torch model아키텍처 선택 및 구현 (Resnet + @)모델 학습 및 하이퍼파라미터 튜닝
Jan. 26 (Mon)	<ul style="list-style-type: none">Model evaluationGrad-CAM	<ul style="list-style-type: none">모델 평가Grad-CAM 실습
Jan. 27 (Tue)	<ul style="list-style-type: none">Github에 프로젝트 업로드프로젝트 발표	

Model evaluation

Three sets

The goal of the modeling

- To make model that predict label of unseen data well
 - So, we separate the dataset according to their role

Dataset 1

For training model

Dataset 2

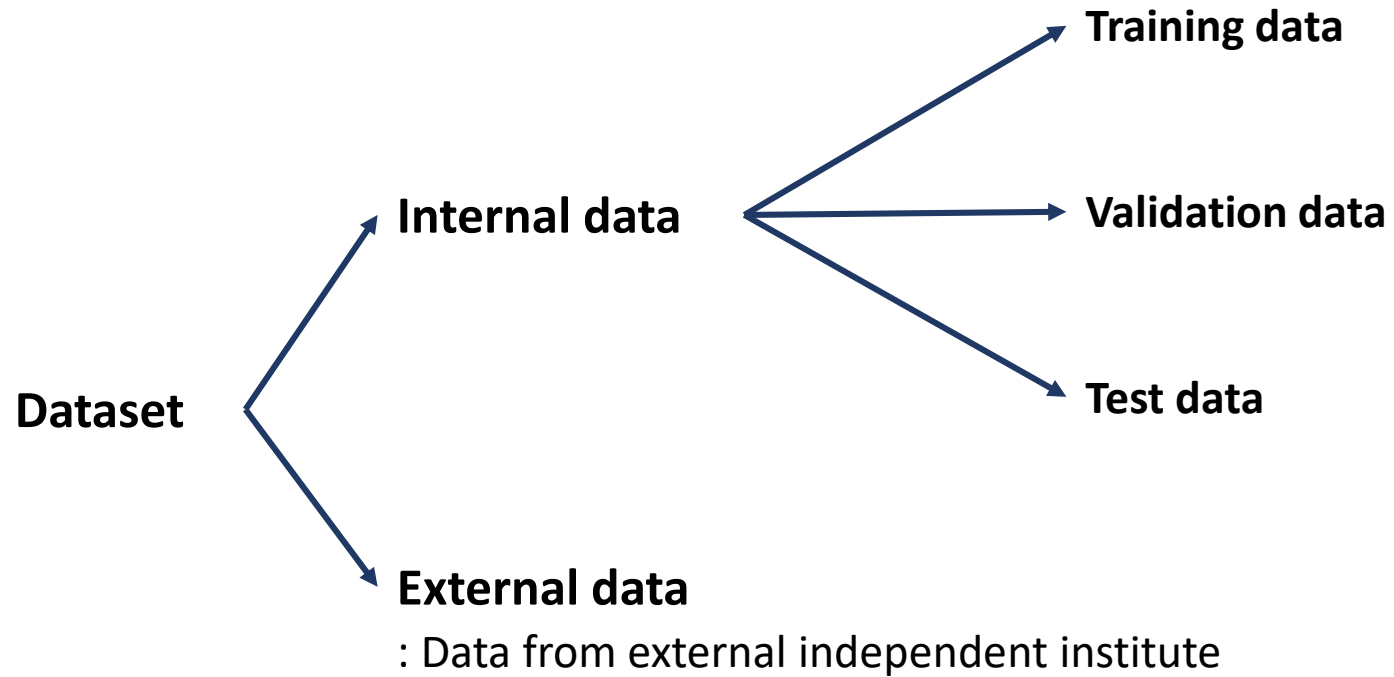
For choosing learning
algorithm and best model
parameter

Dataset 3

For evaluation
performance on unseen
data

Three sets

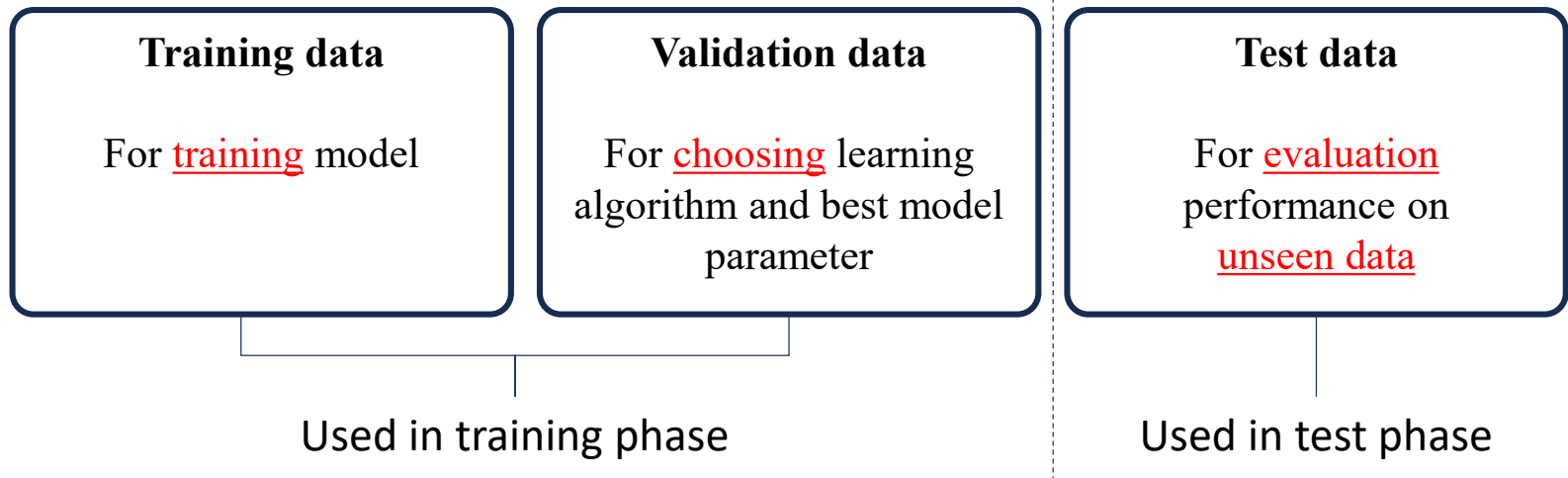
Datasets



Three sets

Three sets

- General classification of dataset



- Training data: largest volume, Test data is also called as hold-out set
- In past, in general, Training : Validation (or Test) = 7 : 3 or 8 : 2
 - Just rule of thumb
 - Ex> For bigdata, 95 : 2.5 : 2.5 also possible

Three sets

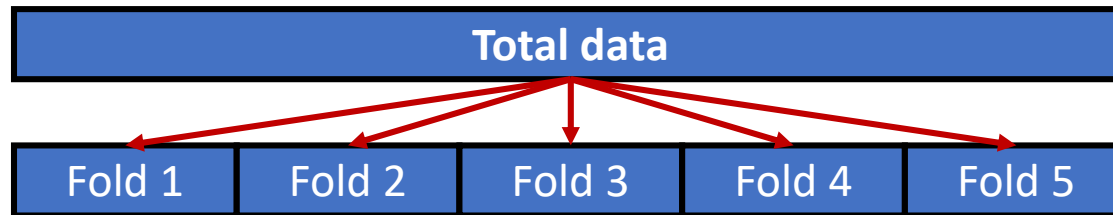
When the amount of data is not enough

- **K-fold cross-validation**
 - Evenly split the data into K part, using each partition as a validation or test set.
 - The divided subsets are called **Fold**.
 - Validate using all the divided data, and the final performance is calculated as the average.

Three sets

When the amount of data is not enough

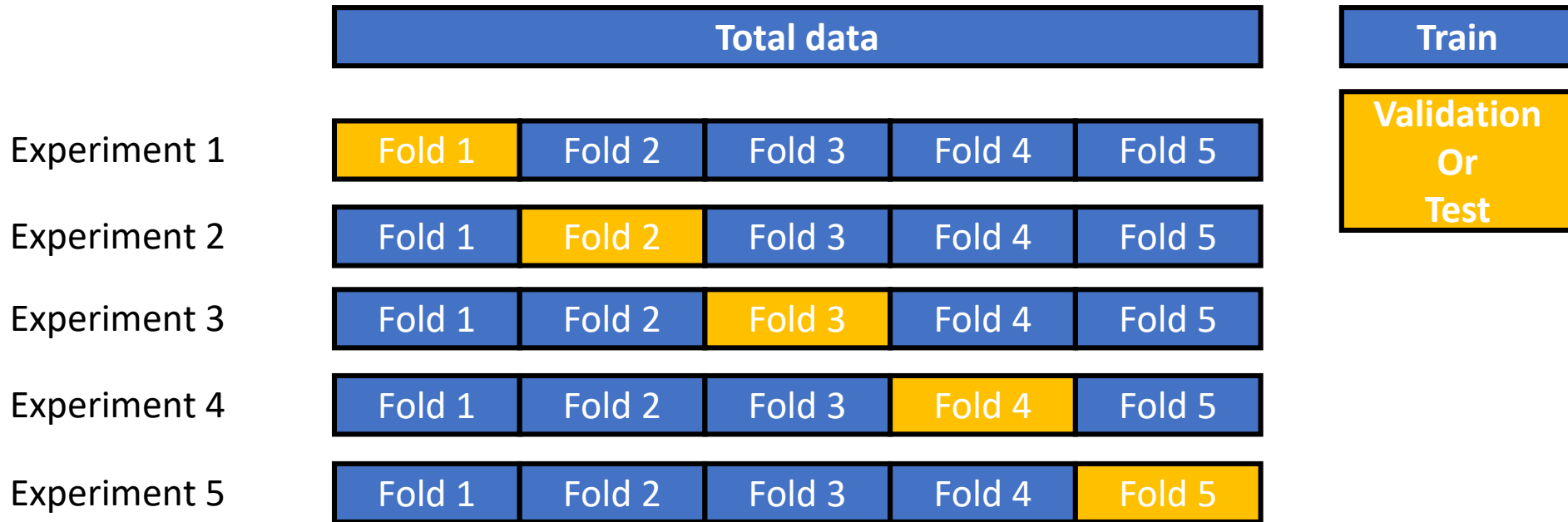
- K-fold cross-validation
 - Ex> 5-fold cross validation



Three sets

When the amount of data is not enough

- K-fold cross-validation
 - Ex> 5-fold cross validation



Underfitting and Overfitting

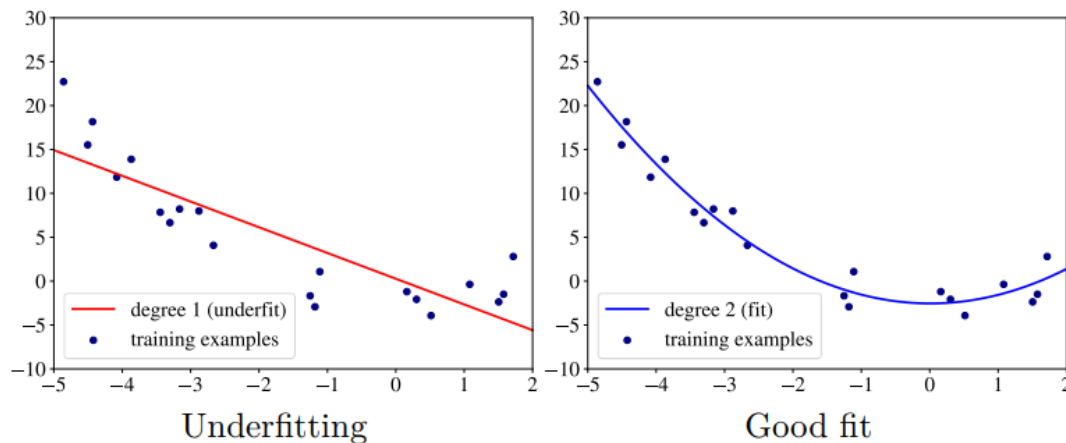
Underfitting

A situation where the **model does not perform well on the training data**.

Potential causes:

1. Too simple model
2. The features that cannot capture meaningful information
3. Insufficient iteration of optimization algorithm

Ex> Too simple model (low complexity)



Underfitting and Overfitting

Underfitting

A situation where the **model does not perform well on the training data.**

Potential solution:

1. Increase model complexity
2. Reconstruct feature set
3. Increase iteration

Underfitting and Overfitting

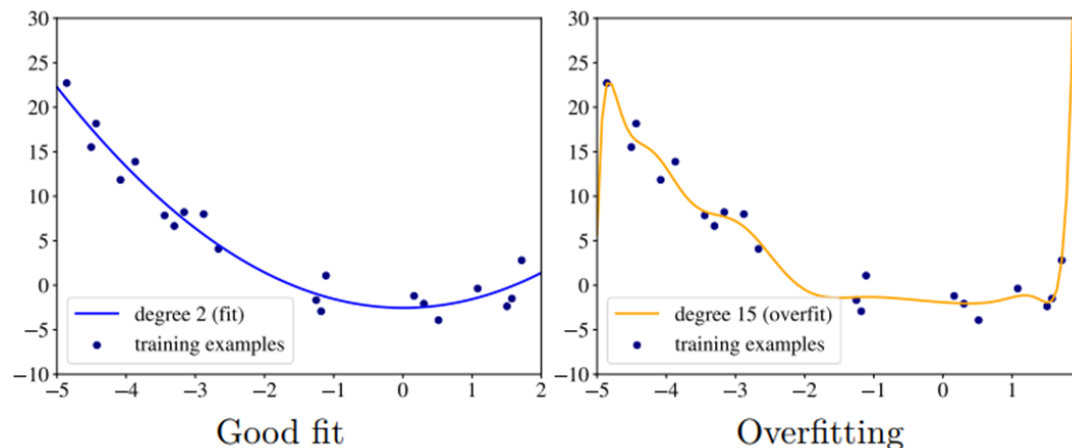
Overfitting

The model **predicts well only on the training data** but **performs poorly on the test data**.

Potential causes:

1. Too much complexity
2. There are too many features relative to the training data.
3. Too much iteration of optimization algorithm

Ex> Too complex model (high complexity)



Underfitting and Overfitting

Overfitting

The model **predicts well only on the training data** but **performs poorly on the test data**.

Potential solution:

1. Simplify model
2. Reduce the size of feature set
3. Reduce iteration of optimization algorithm
4. Add more training data if possible → Increase generalizability

Performance measures for classification

Performance measures

- **Confusion matrix**
- **Accuracy**
- **Precision/Recall**
- **Specificity**
- **Area under the curve (AUC) of receiver operating characteristic (ROC) curve**

Performance measures for classification

Confusion matrix

- Summary of prediction results
- Comparing the actual (true) class labels with the predicted class labels

		Prediction	
		Positive label	Negative label
Ground truth	Positive label	True positive	False negative
	Negative label	False positive	True negative

Performance measures for classification

Confusion matrix

- Summary of prediction results
- Comparing the actual (true) class labels with the predicted class labels

```
from sklearn.metrics import confusion_matrix

y_true = [0, 1, 1, 0, 1, 1, 0, 1, 1]
y_pred = [0, 0, 1, 0, 1, 1, 1, 1, 1]

cm = confusion_matrix(y_true, y_pred)
print(cm)
```

Performance measures for classification

Accuracy

- The proportion of true results (both true positives and true negatives) among the total number of cases.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

		Prediction	
		Positive label	Negative label
Ground truth	Positive label	True positive	False negative
	Negative label	False positive	True negative

Performance measures for classification

Accuracy

- The proportion of true results (both true positives and true negatives) among the total number of cases.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

		Prediction	
		Positive label	Negative label
Ground truth	Positive label	8	12
	Negative label	4	6

$$Accuracy = \frac{8 + 6}{8 + 6 + 4 + 12} = \frac{14}{30}$$

Performance measures for classification

Precision

- The proportion of **true positives among the total predicted positives**

$$Precision = \frac{TP}{TP + FP}$$

		Prediction	
		Positive label	Negative label
Ground truth	Positive label	8	12
	Negative label	4	6

$$Precision = \frac{8}{8 + 4} = \frac{8}{12} = 0.666666 \dots$$

Performance measures for classification

Recall (Sensitivity, True positive rate)

- The proportion of **true positives among the total actual positives**

$$Recall = \frac{TP}{TP + FN}$$

		Prediction	
		Positive label	Negative label
Ground truth	Positive label	8	12
	Negative label	4	6

$$Recall = \frac{8}{8 + 12} = \frac{8}{20} = 0.4$$

Performance measures for classification

Specificity (True negative rate)

- The proportion of **true negatives among the total actual negatives**

$$\text{Specificity} = \frac{TN}{TN + FP}$$

		Prediction	
		Positive label	Negative label
Ground truth	Positive label	8	12
	Negative label	4	6

$$\text{Specificity} = \frac{6}{6 + 4} = \frac{6}{10} = 0.6$$

Performance measures for classification

Area Under the Curve (AUC) of Receiver Operating Characteristic (ROC) Curve

- It can be used with classification models that **output probabilities or scores**.
- By applying a threshold to these probabilities (ranging from 0 to the maximum value), **the True Positive Rate (TPR) and False Positive Rate (FPR) are calculated**.

ROC curve

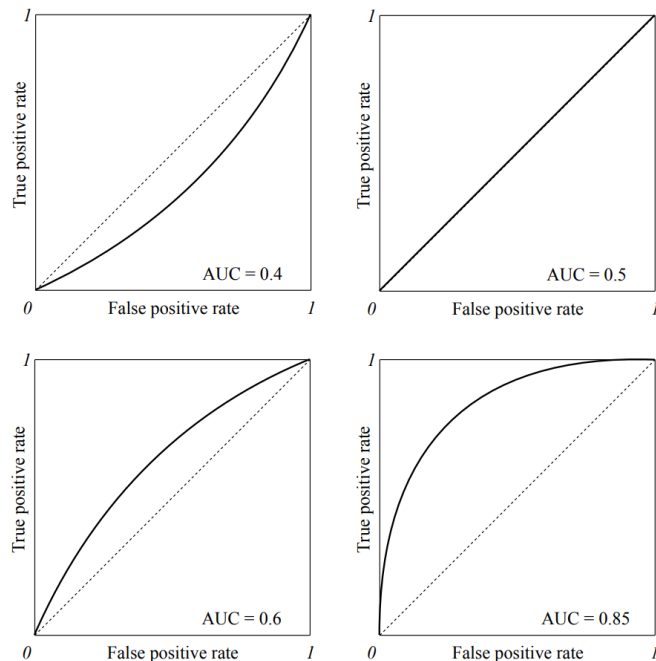


Figure 3: Area under the ROC curve.

- **Random decision**
→ **AUC 0.5**
- **Perfect model**
→ **AUC 1**
- **AUC < 0.5**
→ **Model is inverting prediction**

True Positive Rate (TPR): The proportion of actual positives correctly predicted as positive.

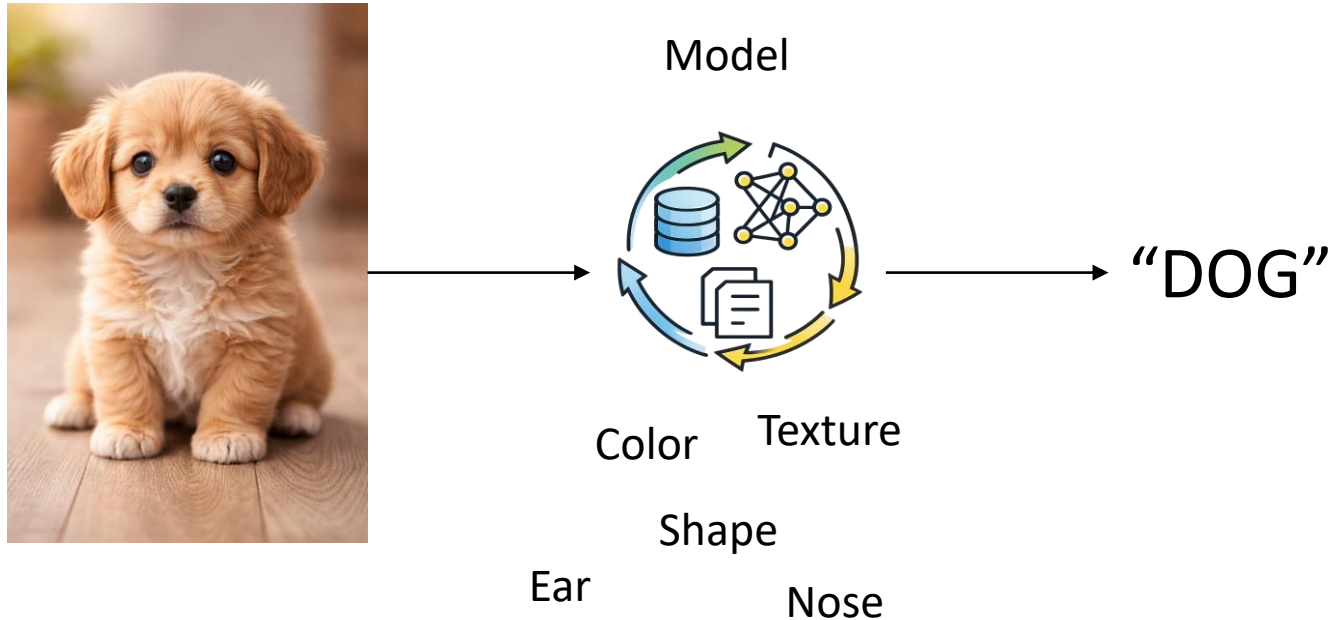
False Positive Rate (FPR): The proportion of actual negatives incorrectly predicted as positive.

Grad-CAM

Grad-CAM

Model interpretability

“Why a model makes a particular prediction, not just what it predicts”

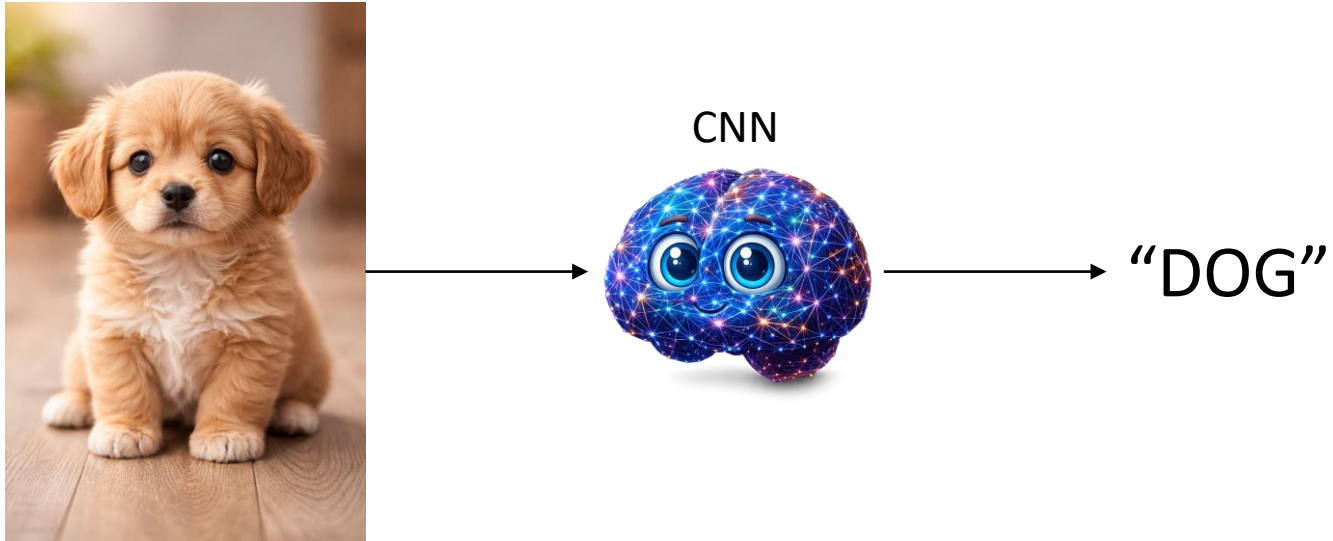


Using interpretable models, we are able to:

- Identify which features influenced the decision
- Understand how changes in input affect the output
- Explain the model's behavior to humans

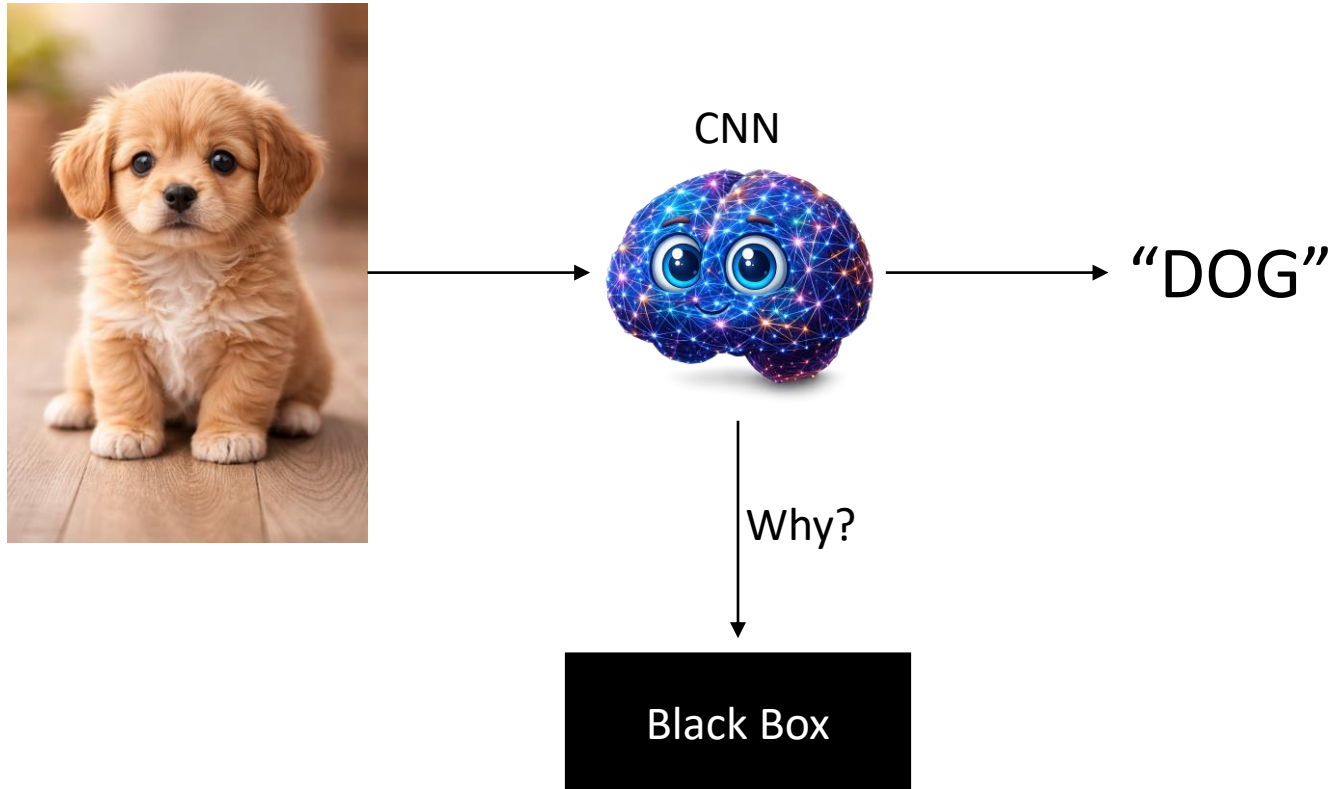
Grad-CAM

Model interpretability



Grad-CAM

Model interpretability



- Deep learning models are not interpretable
→ Instead, we use indirect methods to understand model's behavior

Class activation map (CAM)



This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

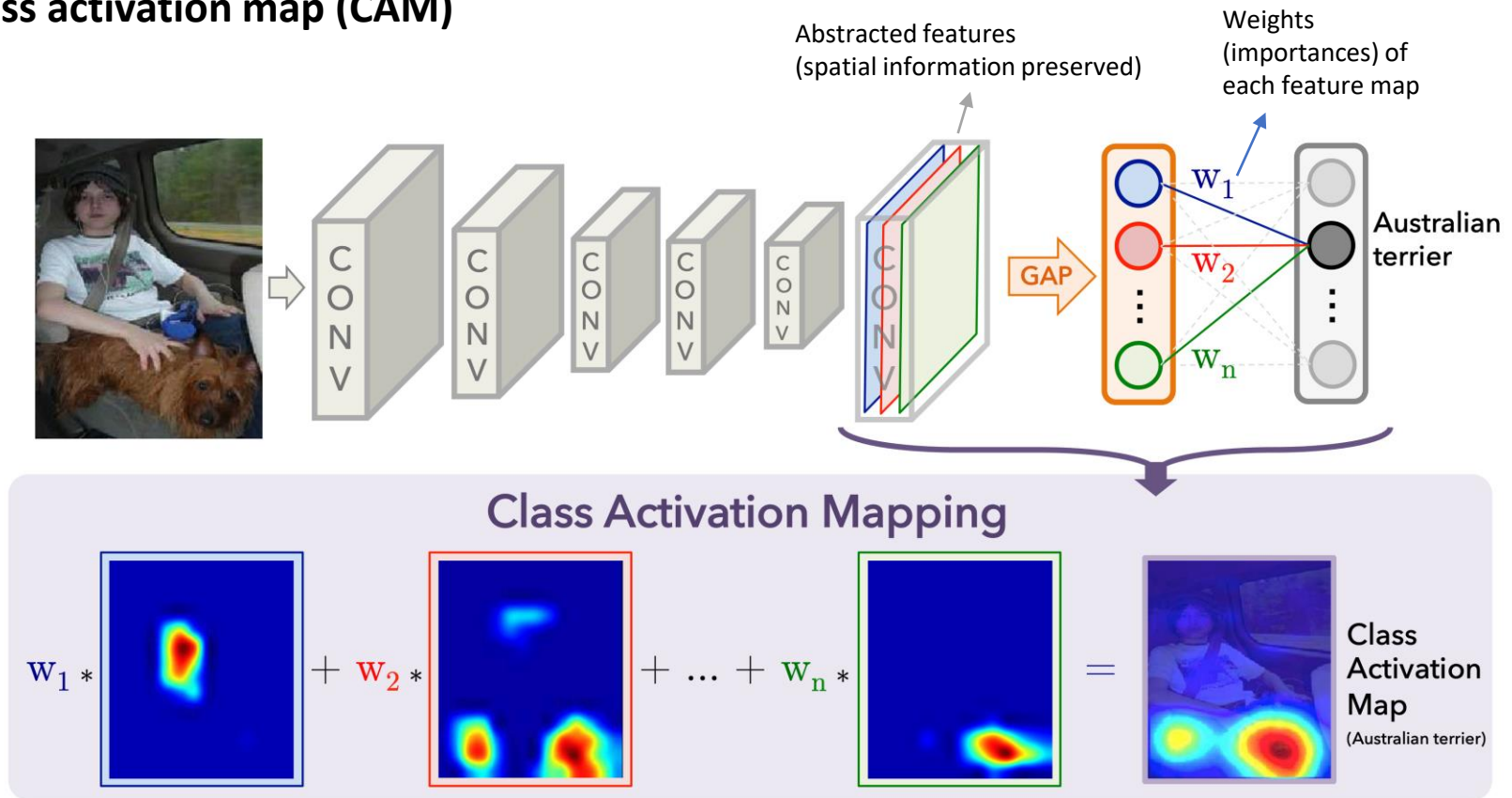
Learning Deep Features for Discriminative Localization

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba
Computer Science and Artificial Intelligence Laboratory, MIT
{bzhou, khosla, agata, oliva, torralba}@csail.mit.edu

- Zhou, Bolei, et al. "Learning deep features for discriminative localization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Grad-CAM

Class activation map (CAM)



Class activation map (CAM)

- When $f_k(x, y)$ is activation of unit k in the last conv layer
 - Global average pooling of unit k :

$$F^k = \sum_{x,y} f_k(x, y)$$

- For a given class c , the input to the softmax S_c (class score):

$$S_c = \sum_k w_k^c F^k = \sum_{x,y} \sum_k w_k^c f_k(x, y)$$

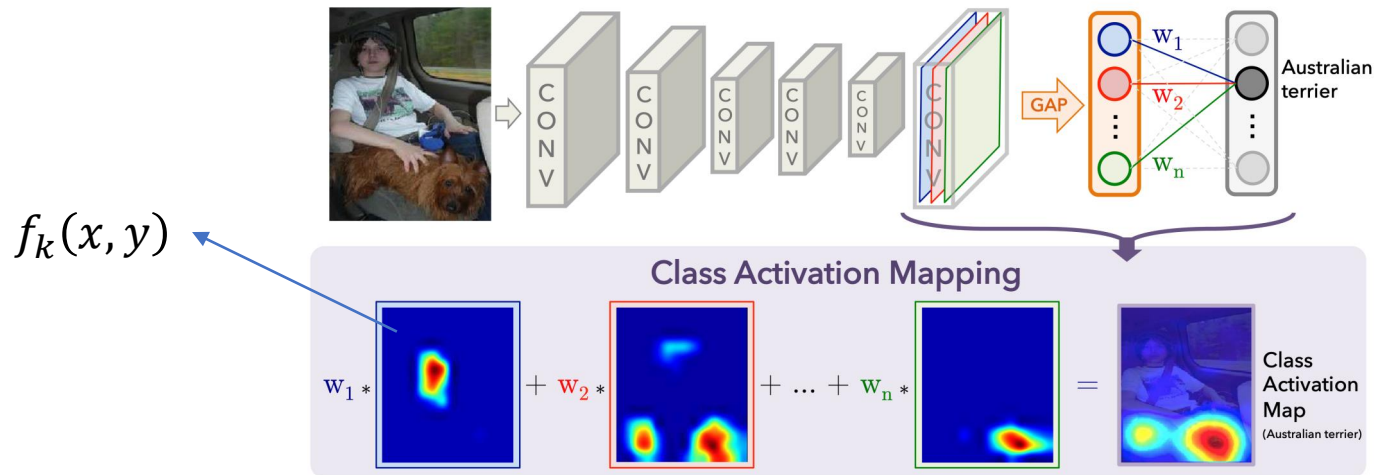
- w_k^c : importance of F^k for class c

Grad-CAM

Class activation map (CAM)

- When $f_k(x, y)$ is activation of unit k in the last conv layer
 - Class activation map for class c ,

$$M_c = \sum_k w_k^c f_k(x, y)$$



Grad-CAM

Class activation map (CAM)

- Example)



Figure 3. The CAMs of two classes from ILSVRC [21]. The maps highlight the discriminative image regions used for image classification, the head of the animal for *briard* and the plates in *barbell*.

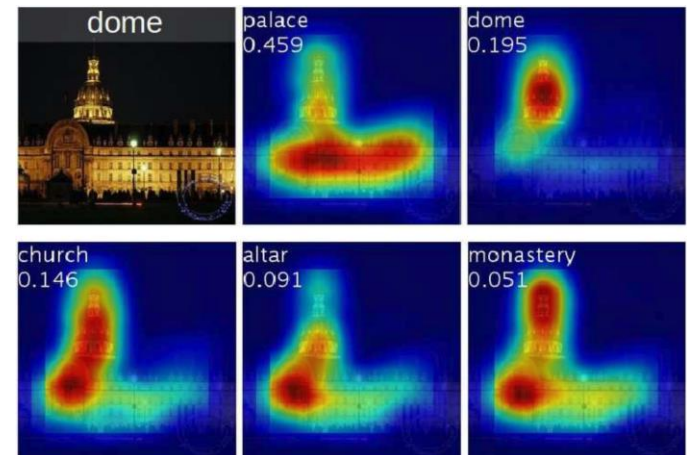


Figure 4. Examples of the CAMs generated from the top 5 predicted categories for the given image with ground-truth as dome. The predicted class and its score are shown above each class activation map. We observe that the highlighted regions vary across predicted classes e.g., *dome* activates the upper round part while *palace* activates the lower flat part of the compound.

Grad-CAM

Class activation map (CAM)

- Example) Localization

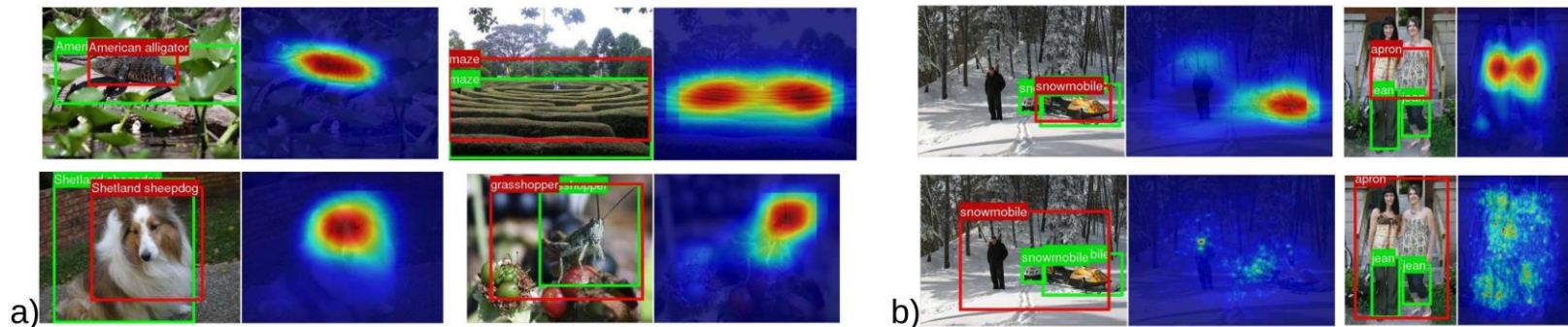


Figure 6. a) Examples of localization from GoogleNet-GAP. b) Comparison of the localization from GoogleNet-GAP (upper two) and the backpropagation using AlexNet (lower two). The ground-truth boxes are in green and the predicted bounding boxes from the class activation map are in red.

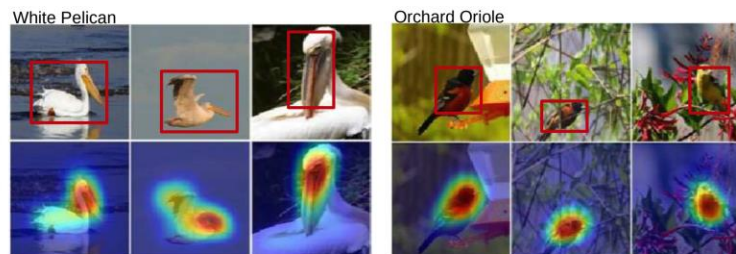


Figure 7. CAMs and the inferred bounding boxes (in red) for selected images from four bird categories in CUB200. In Sec. 4.1 we quantitatively evaluate the quality of the bounding boxes (41.0% accuracy for 0.5 IoU). We find that extracting GoogLeNet-GAP features in these CAM bounding boxes and re-training the SVM improves bird classification accuracy by about 5% (Tbl. 4).

Grad-CAM

Class activation map (CAM)

- Example) VQA

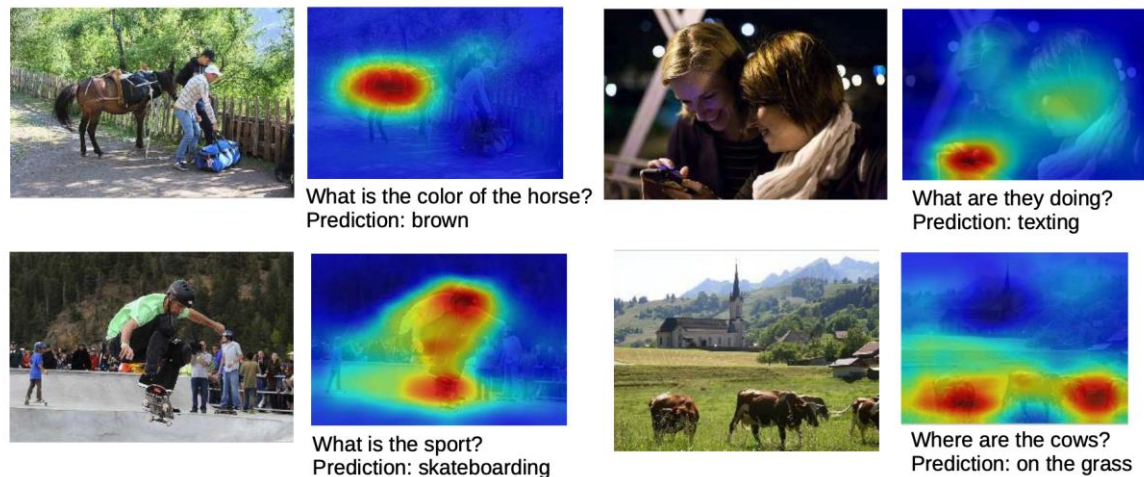
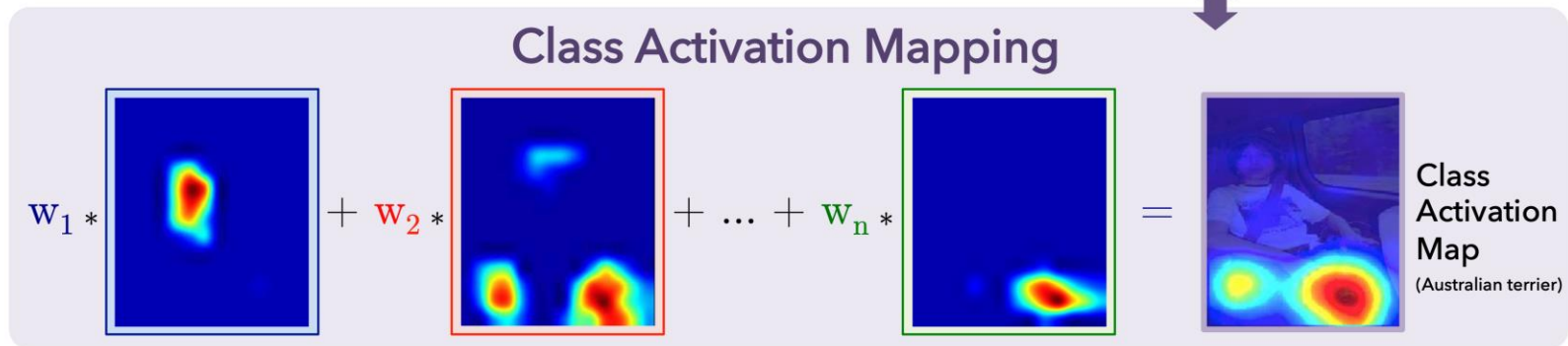
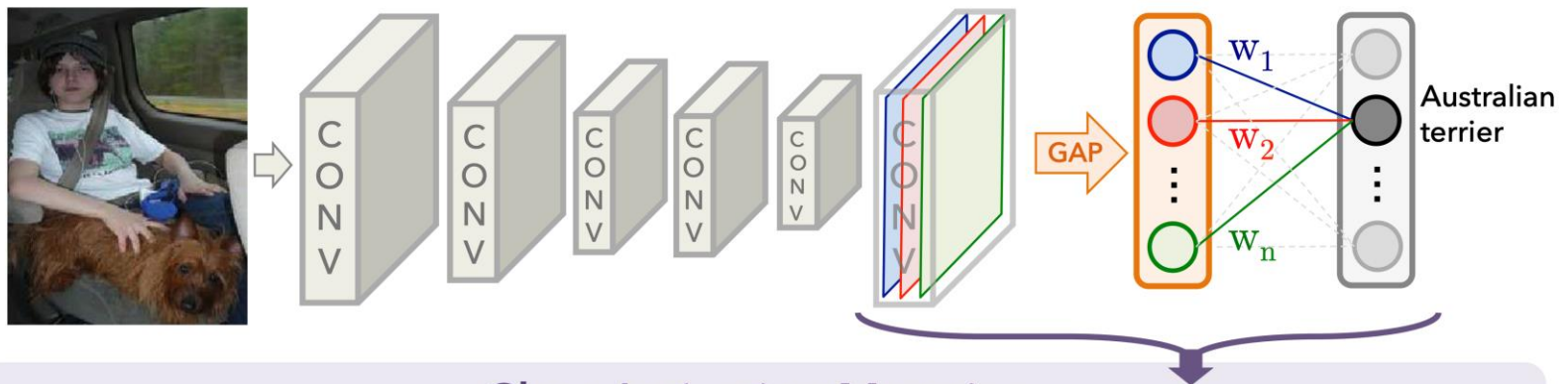


Figure 12. Examples of highlighted image regions for the predicted answer class in the visual question answering.

Grad-CAM

Class activation map (CAM)

- Limitation
 - An architecture should contain [CONV-GAP-Linear mapping] structure



Gradient-weighted Class Activation Map (Grad-CAM)



This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju^{1*} Michael Cogswell¹ Abhishek Das¹ Ramakrishna Vedantam^{1*}
Devi Parikh^{1,2} Dhruv Batra^{1,2}
¹Georgia Institute of Technology ²Facebook AI Research
{ramprs, cogswell, abhshkdz, vrama, parikh, dbatra}@gatech.edu

- Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Grad-CAM

Gradient-weighted Class Activation Map (Grad-CAM)

- Produces a **heatmap** highlighting important regions
- **Uses gradients**
- Architecture-agnostic
- Applicable to pretrained models

Gradient-weighted Class Activation Map (Grad-CAM)

“Interpretability matters”

- In order to build trust in intelligent systems and move towards their meaningful integration into our everyday lives,
 - it is clear that we must build ‘transparent’ models that explain why they predict what they predict.
 - So that, when
 - $AI < Human$: identify the failure modes → thereby helping researchers focus their efforts on the most fruitful research directions
 - $AI = Human$: establish appropriate trust and confidence in users
 - $AI > Human$: Machine teaching (machine teaching a human)

Gradient-weighted Class Activation Map (Grad-CAM)

- Trade-off between accuracy and simplicity (or interpretability)
 - Classical rule-based or expert system
 - Not very accurate (or robust)
 - Hand-designed stages → interpretable
 - Deep model
 - We sacrifice interpretable modules
 - Achieve greater performance using greater abstraction (more layers) and tighter integration (end-to-end training)

Grad-CAM

Gradient-weighted Class Activation Map (Grad-CAM)

- What makes a good visual explanation?
 - For classification, to justify target category,
 - Class discriminative (I.e., localize the category in the image)
 - High-resolution (i.e. capture fine-grained detail)

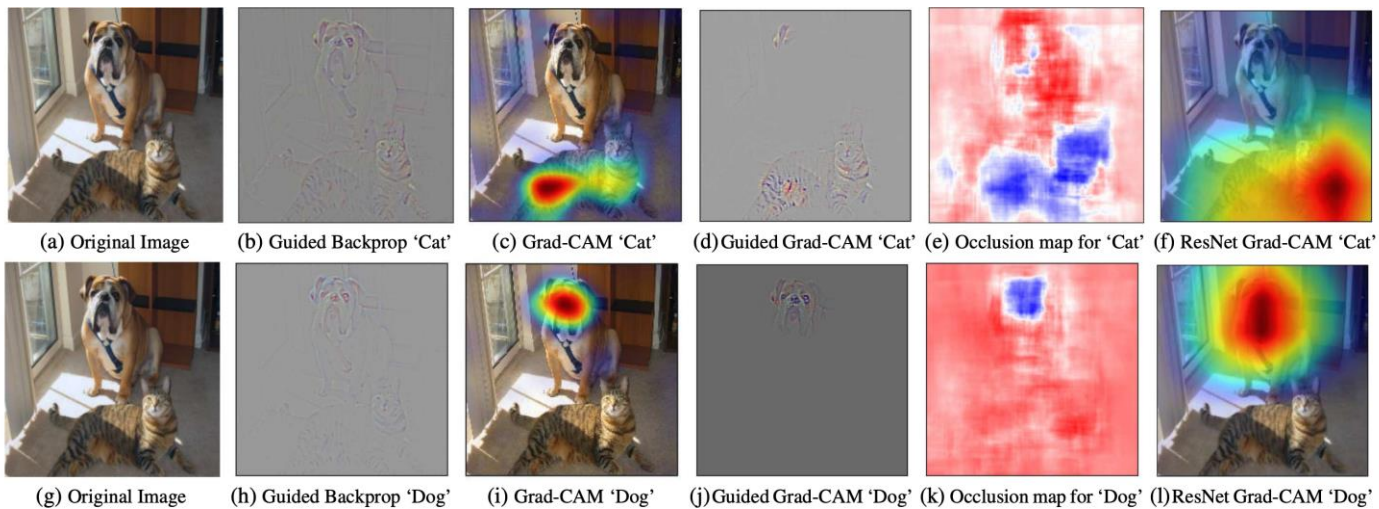


Figure 1: (a) Original image with a cat and a dog. (b-f) Support for the cat category according to various visualizations for VGG-16 and ResNet. (b) Guided Backpropagation [42]: highlights all contributing features. (c, f) Grad-CAM (Ours): localizes class-discriminative regions, (d) Combining (b) and (c) gives Guided Grad-CAM, which gives high-resolution class-discriminative visualizations. Interestingly, the localizations achieved by our Grad-CAM technique, (c) are very similar to results from occlusion sensitivity (e), while being orders of magnitude cheaper to compute. (f, l) are Grad-CAM visualizations for ResNet-18 layer. Note that in (c, f, i, l), red regions corresponds to high score for class, while in (e, k), blue corresponds to evidence for the class. Figure best viewed in color.

Grad-CAM

Gradient-weighted Class Activation Map (Grad-CAM)

- What makes a good visual explanation?
 - Class discriminative (I.e., localize the category in the image)
 - High-resolution (i.e. capture fine-grained detail)

High resolution, but not class discriminative

Highly discriminative

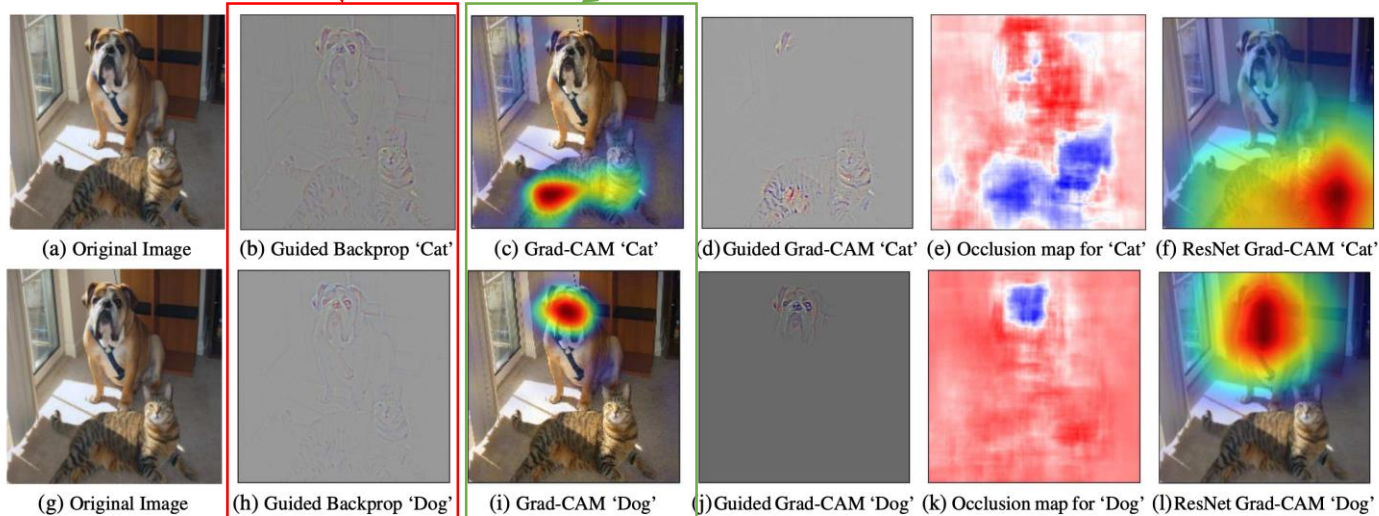


Figure 1: (a) Original image with a cat and a dog. (b-f) Support for the cat category according to various visualizations for VGG-16 and ResNet. (b) Guided Backpropagation [42]: highlights all contributing features. (c, f) Grad-CAM (Ours): localizes class-discriminative regions. (d) Combining (b) and (c) gives Guided Grad-CAM, which gives high-resolution class-discriminative visualizations. Interestingly, the localizations achieved by our Grad-CAM technique, (c) are very similar to results from occlusion sensitivity (e), while being orders of magnitude cheaper to compute. (f, l) are Grad-CAM visualizations for ResNet-18 layer. Note that in (c, f, i, l), red regions corresponds to high score for class, while in (e, k), blue corresponds to evidence for the class. Figure best viewed in color.

Grad-CAM

Gradient-weighted Class Activation Map (Grad-CAM)

- What makes a good visual explanation?
 - Class discriminative (I.e., localize the category in the image)
 - High-resolution (i.e. capture fine-grained detail)

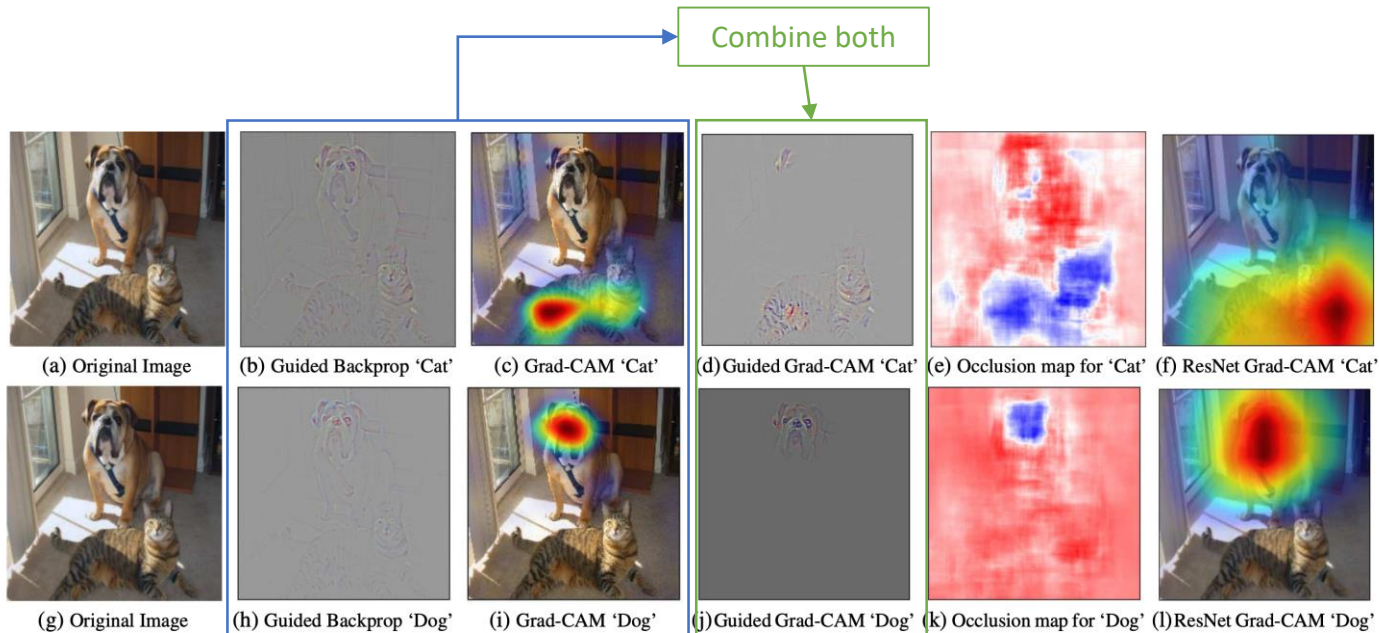


Figure 1: (a) Original image with a cat and a dog. (b-f) Support for the cat category according to various visualizations for VGG-16 and ResNet. (b) Guided Backpropagation [42]: highlights all contributing features. (c, f) Grad-CAM (Ours): localizes class-discriminative regions. (d) Combining (b) and (c) gives Guided Grad-CAM, which gives high-resolution class-discriminative visualizations. Interestingly, the localizations achieved by our Grad-CAM technique, (c) are very similar to results from occlusion sensitivity (e), while being orders of magnitude cheaper to compute. (f, l) are Grad-CAM visualizations for ResNet-18 layer. Note that in (c, f, i, l), red regions corresponds to high score for class, while in (e, k), blue corresponds to evidence for the class. Figure best viewed in color.

Grad-CAM

Gradient-weighted Class Activation Map (Grad-CAM)

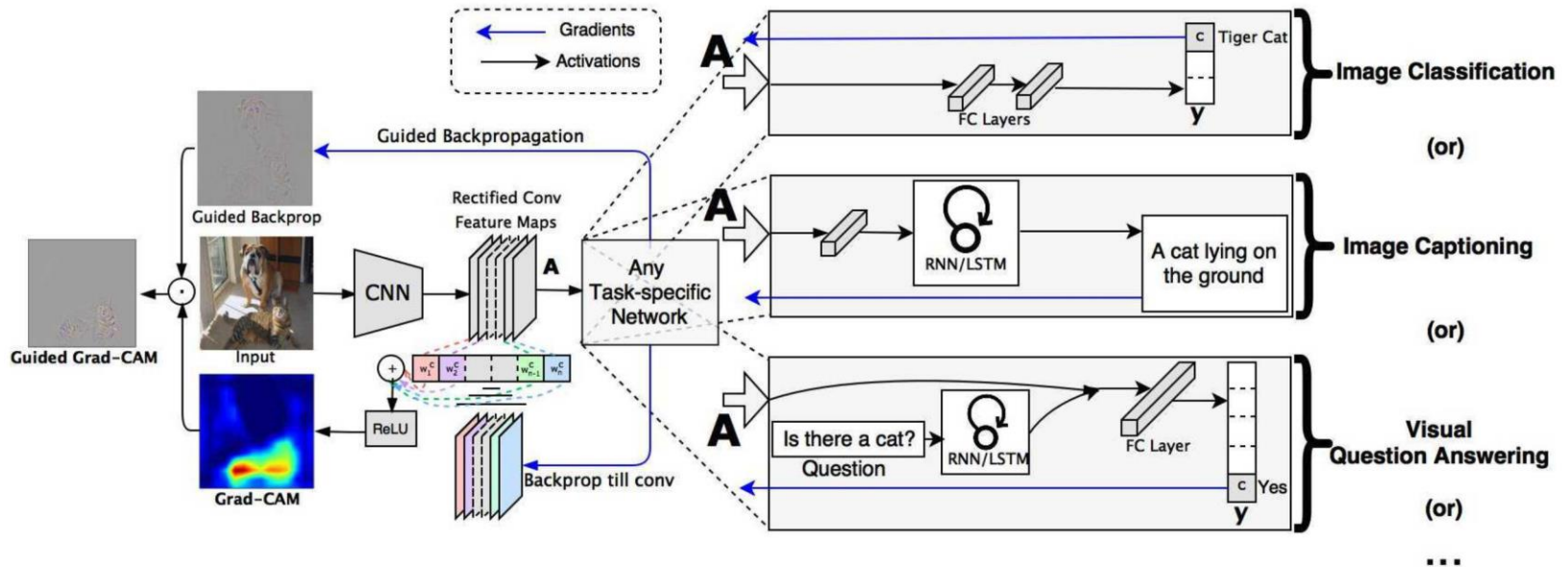


Figure 2: Grad-CAM overview: Given an image and a class of interest (e.g., 'tiger cat' or any other type of differentiable output) as input, we forward propagate the image through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class (tiger cat), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to compute the coarse Grad-CAM localization (blue heatmap) which represents where the model has to look to make the particular decision. Finally, we pointwise multiply the heatmap with guided backpropagation to get Guided Grad-CAM visualizations which are both high-resolution and concept-specific.

Gradient-weighted Class Activation Map (Grad-CAM)

- Deeper representations in a CNN capture higher-level visual constructs
- Convolutional features naturally retain spatial information
→ (lost in FC layer)
 - So we can expect the last convolutional layers to have the best compromise between high-level semantics and detailed spatial information
 - The neurons in these layers look for semantic class-specific information in the image
 - Grad-CAM uses the **gradient information** flowing into the last convolutional layer of the CNN
 - to understand **the importance of each neuron for a decision of interest.**

Grad-CAM

Gradient-weighted Class Activation Map (Grad-CAM)

- At first, we compute the gradient of score for class c , with respect to feature map of a convolutional layer

$$\frac{\partial y^c}{\partial A^k}$$

- Class c , class score y^c , feature map activation A^k
- Neuron importance weight

$$\alpha_k^c = \overbrace{\frac{1}{Z} \sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \quad (1)$$

- Represents a partial linearization of the deep network downstream from A , and captures the 'importance' of feature map k for a target class c .

Gradient-weighted Class Activation Map (Grad-CAM)

- Grad-CAM

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\underbrace{\sum_k \alpha_k^c A^k}_{\text{linear combination}} \right) \quad (2)$$

- This results in a coarse heat-map of the same size as the convolutional feature maps → Resizing required
- ReLU → we are only **interested in the features that have a positive influence** on the class of interest
 - Negative pixels are likely to belong to other categories in the image.
- y^c need not to be class score produce by image classification CNN
 - It could be any differentiable activation

Grad-CAM

Gradient-weighted Class Activation Map (Grad-CAM)

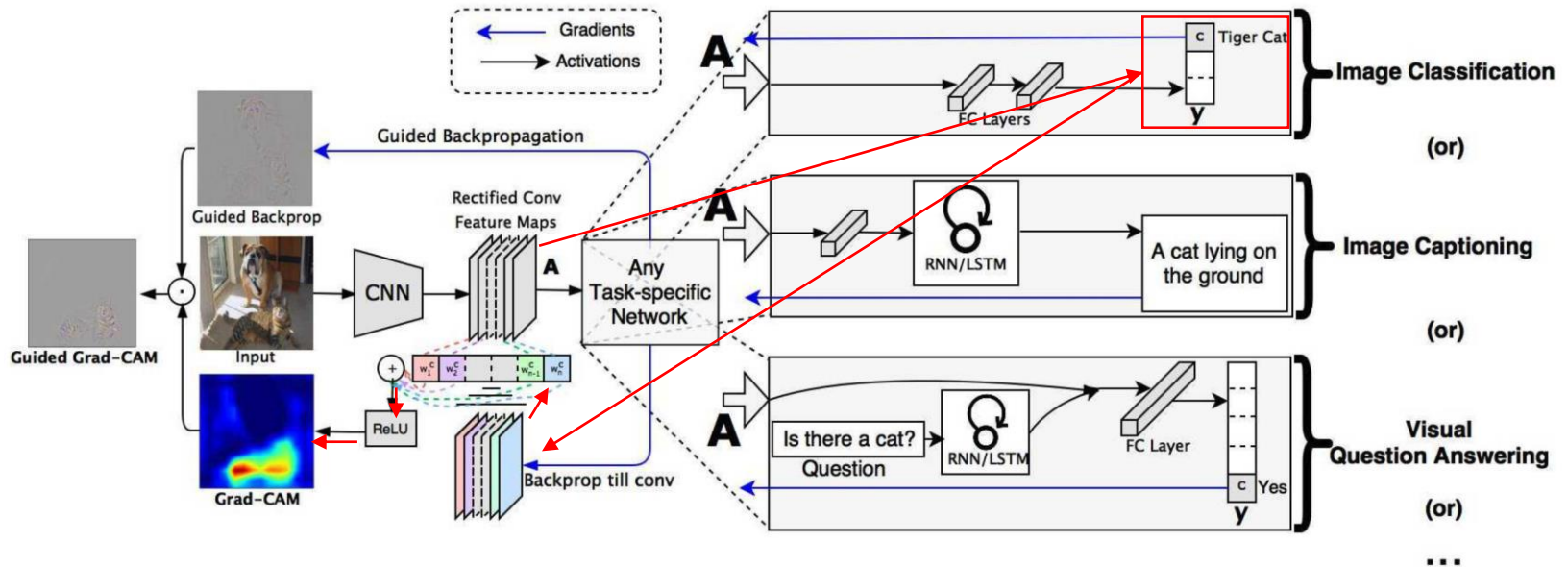


Figure 2: Grad-CAM overview: Given an image and a class of interest (e.g., 'tiger cat' or any other type of differentiable output) as input, we forward propagate the image through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class (tiger cat), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to compute the coarse Grad-CAM localization (blue heatmap) which represents where the model has to look to make the particular decision. Finally, we pointwise multiply the heatmap with guided backpropagation to get Guided Grad-CAM visualizations which are both high-resolution and concept-specific.

Gradient-weighted Class Activation Map (Grad-CAM)

- Example) Failure mode analysis

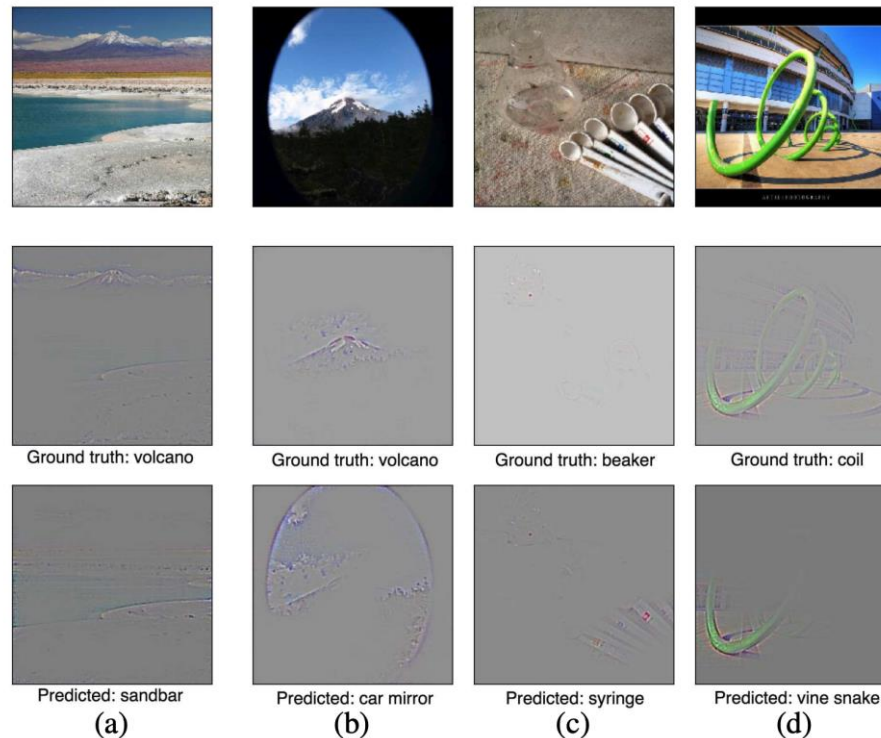


Figure 4: In these cases the model (VGG-16) failed to predict the correct class in its top 1 (a and d) and top 5 (b and c) predictions. Humans would find it hard to explain some of these predictions without looking at the visualization for the predicted class. But with Grad-CAM, these mistakes seem justifiable.

Gradient-weighted Class Activation Map (Grad-CAM)

- Example) VQA explainability

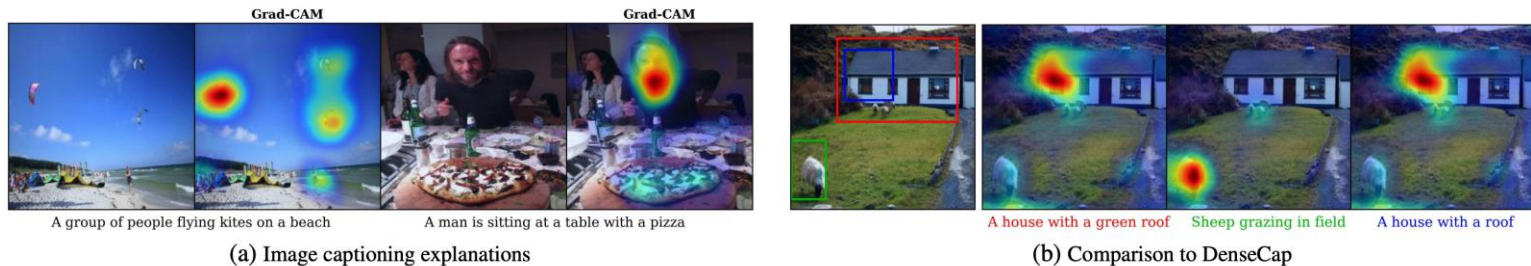
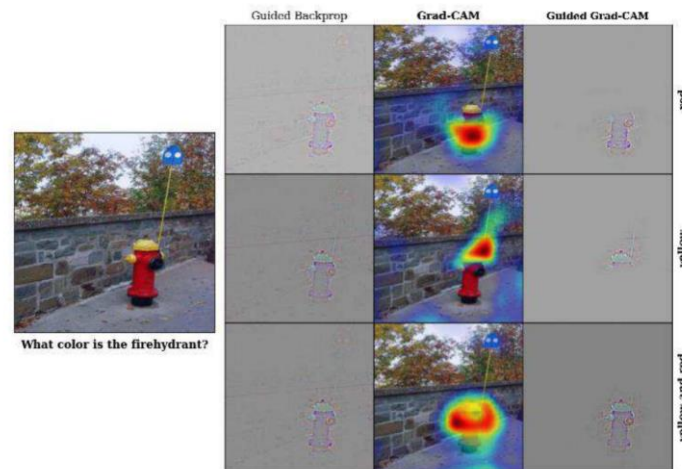


Figure 5: Interpreting image captioning models: We use our class-discriminative localization technique, Grad-CAM to find spatial support regions for captions in images. Fig. 5a Visual explanations from image captioning model [23] highlighting image regions considered to be important for producing the captions. Fig. 5b Grad-CAM localizations of a *global* or *holistic* captioning model for captions generated by a dense captioning model [21] for the three bounding box proposals marked on the left. We can see that we get back Grad-CAM localizations (right) that agree with those bounding boxes – even though the captioning model and Grad-CAM techniques do not use any bounding box annotations.

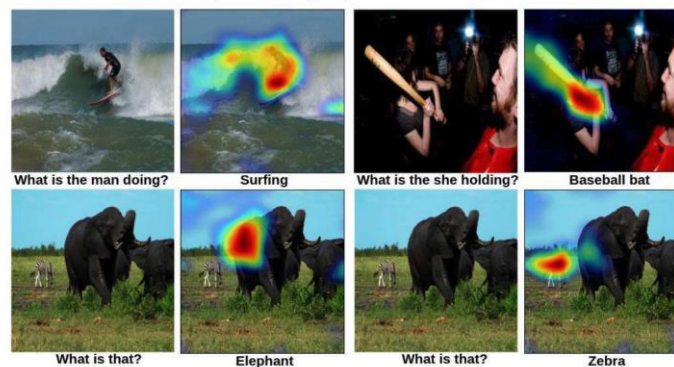
Grad-CAM

Gradient-weighted Class Activation Map (Grad-CAM)

- Example) VQA explainability



(a) Visualizing VQA model from [28]



(b) Visualizing ResNet based Hierarchical co-attention VQA model from [29]