FLEX: Faithful Linguistic Explanations for Neural Net Based Model Decisions

Sandareka Wickramanayake, Wynne Hsu, Mong Li Lee
School of Computing, National University of Singapore
{sandaw, whsu, leeml}@comp.nus.edu.sg

Abstract

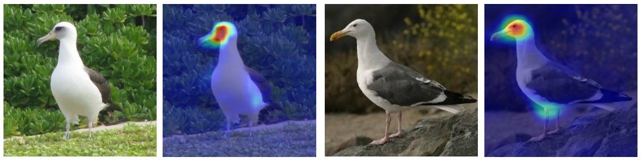
Explaining the decisions of a Deep Learning Network is imperative to safeguard end-user trust. Such explanations must be intuitive, descriptive, and faithfully explain why a model makes its decisions. In this work, we propose a framework called FLEX (Faithful Linguistic EXplanations) that generates post-hoc linguistic justifications to rationalize the decision of a Convolutional Neural Network. FLEX explains a model's decision in terms of features that are responsible for the decision. We derive a novel way to associate such features to words, and introduce a new decision-relevance metric that measures the faithfulness of an explanation to a model's reasoning. Experiment results on two benchmark datasets demonstrate that the proposed framework can generate discriminative and faithful explanations compared to state-of-the-art explanation generators. We also show how FLEX can generate explanations for images of unseen classes as well as automatically annotate objects in images.

Introduction

Despite the remarkable performance of Deep Neural Network (DNN) in many applications, its opaque nature has hindered its usability in the real world (Caruana et al. 2015). Post-hoc interpretation techniques such as *visualizations* and *linguistic descriptions* aim to provide explanations for DNN model decisions (Ribeiro, Singh, and Guestrin 2016).

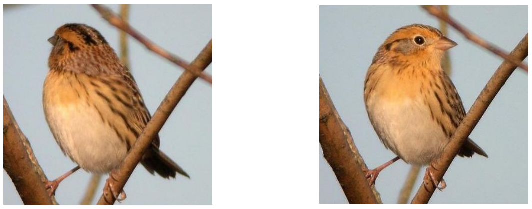
Visualizations indicate which regions/pixels of the input have significantly influenced a model's decision. Figure 1 shows GradCAM (Selvaraju et al. 2016) visual explanations for the decisions of fine-grained bird classification model in (Gao et al. 2016). GradCAM reveals that for these images, it is the region around the bird's head that has influenced the predicted bird categories the most. However, these visualizations do not reveal which aspects of the bird's head region (be it the beak's shape or color, eye ring, or eye color) are responsible for differentiating the different species of birds.

On the other hand, linguistic descriptions such as *"The bird in Figure 1(a) is a Laysan Albatross because of its long thick hooked bill, and the dark shading on its face."* provide specific details which are important for understanding the classification especially when variations among the different classes are subtle. Hendricks et al. (Hendricks et al. 2016) and Park et al. (Park et al. 2018) take into consideration the class labels to generate linguistic descriptions. They adapt a neural image captioning model and condition the model not only on image features but also on the predicted class and train their language generator using the visual descriptions provided by human. However, the explanations generated by their approach may not necessarily reflect the model's actual reasoning (Kim et al. 2017). This is evident as their generated explanation for an image may mention features that are not even visible in the image. Figure 2 shows two birds with different poses. The same explanation is generated by Hendrick's model for these two birds even though black streaks are not visible on the crown of the bird in Figure 2(b).



(a) Laysan Albatross (b) California Gull

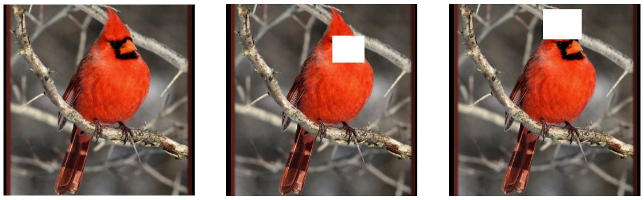
Figure 1: Saliency maps of sample images from CUB.



(a) (b)

Figure 2: Same explanation *"This is an Le Conte Sparrow because this is a small brown bird with black streaks on its crown"* for different images irrespective of whether the features are visible.

In addition, we observe that a model may make its decision using features different from that used by human. Figure 3(a) shows an image of a 'Cardinal'. The feature that human uses to distinguish this type of bird is its "black cheek patch" (Reed et al. 2016). The model's confidence that the bird in Figure 3(a) belongs to the class ‘Cardinal’ is 0.9583. Figure 3(b) and (c) show the model’s confidence when we occlude the black cheek patch and the crown of the bird respectively. There was a greater decrease in confidence when the crown was occluded indicating the model has based its decision more on the feature “crown” than the “black cheek-patch”, unlike the human. Knowing the true rationale behind a model’s decision is crucial particularly in mission-critical applications such as the medical domain.



(a) Conf = 0.9583 (b) Conf = 0.7742 (c) Conf = 0.6731

Figure 3: Model’s confidence that the bird is ‘Cardinal’ when different parts are occluded.

In this work, we propose a novel linguistic justification generator framework called FLEX to elucidate image classification models by generating intuitive and descriptive explanations that truly reflect how the models make their decisions. We justify each classification decision with decision-relevant features that are automatically derived from the model being interpreted and generate a description which is aligned with the model’s decision process.

Inspired by the work in (Selvaraju et al. 2016), we use the gradient of the predicted class backpropagated to the penultimate layer to identify features that contribute more towards the model decision. Then we condition a Long-Short Term Memory based (Hochreiter and Schmidhuber 1997) language generator on these features to generate descriptive explanations. We utilize a relevance loss function to assist the justification generator in picking up words related to visual features that are relevant to the model decision.

We derive a novel way to associate visual features to words. Finding this association frees us from being limited to generate explanations only for classes that FLEX is trained for. This enables FLEX to generate explanations for images of unseen classes where there are no ground truth descriptions. This association allows objects in images to be annotated automatically, thus eliminating the bottleneck in semantic segmentation due to the lack of annotated images.

We evaluate the effectiveness of the proposed framework on CUB (Wah et al. 2011) and MPII (Andriluka et al. 2014) datasets. We introduce a new metric to measure how much the justification matches the model’s reasoning. Experiment results show that our framework can generate discriminative and decision-relevant explanations compared to state-of-art linguistic explanation models.

Related Work

In this section, we review works that provide explanations in the form of visualization and linguistic description.

A visualization technique that uncovers image features such as color, shape, texture, etc. at any inner layer in a CNN model was introduced in (Zeiler and Fergus 2014). They used “decovnet” technique to project stimulation of each feature map back to the input space. The guided backpropagation introduced in (Springenberg et al. 2015) extends this idea by modifying the backward flow of gradients through ReLU (Rectified Linear Units) layers. Saliency map generation techniques (Simonyan, Vedaldi, and Zisserman 2014; Bach et al. 2015) were proposed to highlight regions in the input space that have influenced the model decision most. Grad-CAM (Selvaraju et al. 2016) computes the importance of feature maps of the last convolutional layer towards the predicted class and use weighted feature maps to generate a heatmap which highlights the class-discriminative regions of the image. All these techniques do not provide explicit explanations on *why a model makes a certain decision*.

Recent approaches try to use more descriptive linguistic explanations to explain CNN model decisions. These include a mechanism for interpretable models (Barratt 2017), post-hoc interpretation technique in terms of discriminative features (Hendricks et al. 2016), and an attention-based multi-modal explanation framework (Park et al. 2018). The reinforcement learning based discriminative loss in (Hendricks et al. 2016) enforced class specificity and conditioned the language generator not only on the image but also on the predicted class. However, directly conditioning on the class label may lead to the generation of the same explanation for images belonging to the same class regardless of the applicability of the explanation for a target image. The explainer in (Park et al. 2018) used visual features weighted by an attention map to generate explanations. However, the attention map is generated solely based on the class label and does not involve in the decision-making process of the classifier. Thus, the multi-modal explanations may not capture the true visual features used by the classifier.

FLEX Framework

The proposed framework has three key steps. We first identify features that contribute to a model’s decision by extracting a subset of important feature maps from the CNN classification model. Next, we associate words to these features so that we can derive words describing decision-relevant features. Finally, the weighted image features are fed into a justification generator based on LSTM units. We introduce a relevance loss function which is based on the identified decision-relevant words to ensure that the generated justifications faithfully describe how the model makes its decision. Figure 4 gives an overview of FLEX.

Identify Important Features

The contribution of hidden units in a CNN towards its decisions can be identified in different granularities such as per neuron, per channel (filter response also known as feature map) or per a factorization of the whole layer (Olah et al. 2018). Given an image and its predicted class, we identify the importance of each feature map from various convolutional layers.

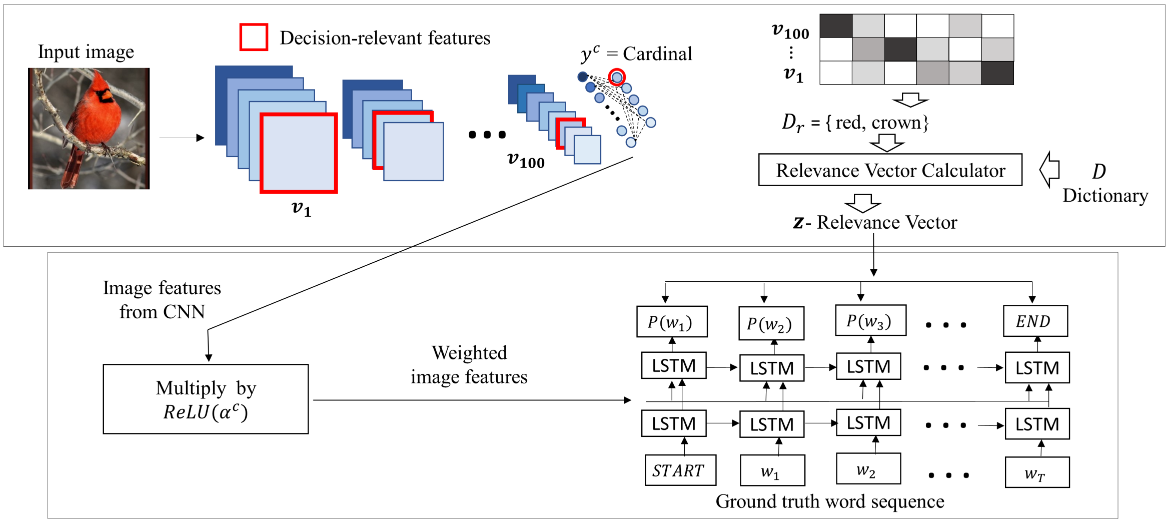


Figure 4: Overview of FLEX Framework.

Suppose we have a CNN that classifies an image I into some class c. Let $\{\boldsymbol{A^1}, \boldsymbol{A^2}, ..., \boldsymbol{A^K}\}$ be the set of feature maps at some layer h in CNN. In order to determine the importance of a feature map $\boldsymbol{A}$, we compute the gradient of the score y^c for class c with respect to each ij^{th} neuron of the feature map $\boldsymbol{A}$, denoted by a_{ij}, and take the global average. Thus, the weight α^c of feature map $\boldsymbol{A}$ with respect to c is given by:

$$\alpha^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial a_{ij}} \tag{1}$$

where Z is the total number of neurons in $\boldsymbol{A}$.

We apply softmax to the weight α^c to obtain the normalized score β^c:

$$\beta^c = \frac{f(\alpha^c)}{\sum f(\alpha^c)} \quad \text{where} \quad f(x) = e^x \tag{2}$$

We sort the feature maps according to their scores and select a minimum set of feature maps $\boldsymbol{A_h}$ such that $\sum \beta^c$ is greater than some given threshold η. This threshold is determined based on the percentage of contribution we want to consider in selecting feature maps from each layer. In our experiments, we set η to 80%. This process is repeated for each layer in the CNN and we obtain the set of important feature maps F for all the convolutional layers.

Associate Words to Features

Next, we want to associate some word w to each important feature $\boldsymbol{v} \in F$ such that w best describes $\boldsymbol{v}$.

Suppose we have a training dataset of images, their class labels, and ground truth descriptions. Let D_I be the ground truth description for each image I. Then we have a word dictionary $D = \bigcup D_I$. For each noun/adjective $w \in D$, we compute its co-occurrence score with each feature $\boldsymbol{v} \in F$ using the Dice's coefficient in Equation 3:

$$score(w, \boldsymbol{v}) = \frac{2 \times occur(w, \boldsymbol{v})}{count(w) + count(\boldsymbol{v})} \tag{3}$$

where $count(w)$ is the number of occurrences of w in the ground truth descriptions, $count(\boldsymbol{v})$ is the number of occurrences of $\boldsymbol{v}$ in the training set, and $occur(w, \boldsymbol{v})$ is the number of times v and w occur together.

The word w with the highest co-occurrence score is associated with the feature $\boldsymbol{v}$. Figure 5 shows that the words 'belly' and 'white' are associated with the features $\boldsymbol{v'}$ and $\boldsymbol{v}$" respectively.

Describe Decision-Relevant Features

Features at different layers represent concepts at different granularities. For example, a feature at the last layer of a CNN may represent the concept of "ear", while the middle layer feature may represent the concept of "flurry". In order to generate coherent linguistic justification, we start from the last convolutional layer L to get the top-k features based on their importance scores and their associated words. These words depict the concepts used by the CNN model to make its classification decision c for a given image I.

For each of these concepts, we obtain its finer granularity description by computing the gradient of activations with respect to the feature maps in layer $L - 1$, and obtain the top-k features as well as their associated words. We do this recursively until we reach the first convolutional layer of the network. The union of all the words associated with the top-k features from layer 1 to L forms the set of words D_r that describe the features used by the model for its decision.

We compute the relevance of a word w to the set D_r as follows:

$$\text{relevance}(w, D_r) = \begin{cases} 0 & w \notin D_r \\ -log\left(\frac{|D_r|}{|D|}\right) & \text{otherwise} \end{cases}$$

Similarly, we can compute the relevance of a word w to the ground truth description of an image I as:

$$\text{relevance}(w, D_I) = \begin{cases} 0 & w \notin D_I \\ -log\left(\frac{|D_I|}{|D|}\right) & \text{otherwise} \end{cases}$$

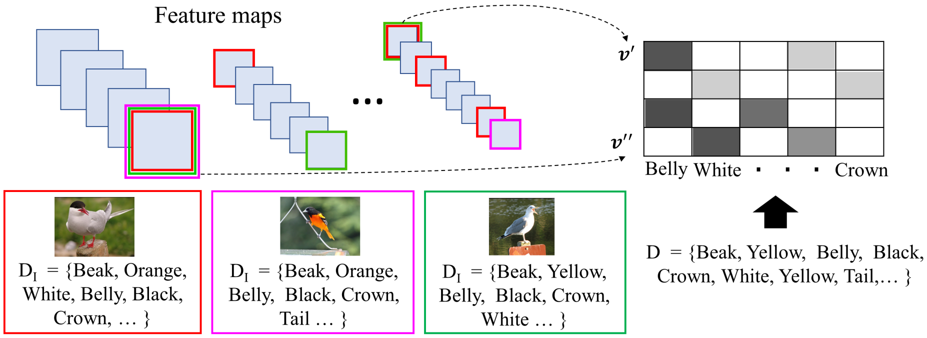

Feature maps
v'
v''
Belly White · · · Crown
D = {Beak, Yellow, Belly, Black, Crown, White, Yellow, Tail,… }
D_I = {Beak, Orange, White, Belly, Black, Crown, … }
D_I = {Beak, Orange, Belly, Black, Crown, Tail … }
D_I = {Beak, Yellow, Belly, Black, Crown, White … }

Figure 5: Associating words to filter responses.

Finally, let $D_c = \bigcup D_I$ where class label of I is c. Then the relevance of a word w to D_c, relevance(w, D_c), can be similarly determined as above.

Given the dictionary of words D, we construct a relevance vector $\boldsymbol{z} = (g(w_1), g(w_2), ..., g(w_{|D|}))$ where w_i is the i^{th} word in D and

$$g(w_i) \quad = \quad \text{relevance}(w_i, D_r) + \text{relevance}(w_i, D_I) + \text{relevance}(w_i, D_c)$$

For example, the set of words relevant to the class "Cardinal" is $D_c = \{red, crown, black, cheekpatch, gray, wing, belly, beak\}$. If we have the image I as shown in Figure 3(a), then $D_I = \{red, crown, black, cheekpatch, belly, beak\}$. The set of descriptive words for features that a CNN model may use to classify the bird in Figure 3(a) as a Cardinal is $D_r = \{red, crown\}$.

Generate Linguistic Justification

Finally, FLEX generates linguistic justifications based on two stacked LSTMs. The training for the LSTMs is as follows. For each input image in the training dataset, we obtain visual features from the penultimate layer of the CNN and calculate their importance towards model decision as in Equation 1. To eliminate features with negative influence, ReLU is applied to the importance scores. We weight visual features with their refined importance and create visual feature vector $\boldsymbol{V}$. A training instance consists of this visual feature vector $\boldsymbol{V}$, ground truth description of the image comprising of a sequence of T words $<w_1, ..., w_T>$, and the corresponding relevance vector $\boldsymbol{z}$ as described earlier.

The first LSTM takes the ground truth word w_{t-1} and its hidden state $\boldsymbol{s^1_{t-1}}$ at time step $t-1$ as inputs and compute the next state, $\boldsymbol{s^1_t}$. Then we concatenate $\boldsymbol{s^1_t}$ with the visual feature vector $\boldsymbol{V}$. This becomes the input to the second LSTM which outputs $\boldsymbol{s^2_t}$. We encode $\boldsymbol{s^2_t}$ to the vocabulary space to produce the conditional probability distribution $P(w_{t-1}|w_{\leq t-1}, I)$ which is used to sample the current word w_t.

Together with the relevance vector $\boldsymbol{z}$, we carry out an element-wise multiplication to compute the relevance loss, denoted as $loss(w_t, I)$, incurred by the justification generator by predicting the word w_t.

$$loss(w_t, I) = max(\boldsymbol{z} \odot P(w_t|w_{\leq t-1}, I)) \tag{4}$$

Our objective function is a linear combination of negative log-likelihood and relevance loss with regularization λ:

$$\sum_{t=1}^{T} [-\log P(w_t|w_{\leq t-1}, I) - \lambda\, loss(w_t, I)] \tag{5}$$

During the inference, we sample from the conditional distribution to get the next word in the generated justification and provide it as the next input to the LSTM.

Experimental Study

In this section, we carry out experiments to compare our framework with the state-of-the-art linguistic explanation frameworks, visual explanations (GVE) (Hendricks et al. 2016) and multi-modal explanations (MME) (Park et al. 2018). We also implemented a baseline model which assumes the weights of all visual features to be 1, and does not include the relevance loss in its objective function.

We use the following datasets for our experiments.

- **CUB.** This is the Caltech UCSD Birds dataset containing 11,788 images of birds belonging to 200 classes (Wah et al. 2011). Each image is annotated with 15 object parts and has 5 sentences describing the details of the bird species (Reed et al. 2016).
- **MPII.** This dataset has 25k images of human poses for different activities (Andriluka et al. 2014). There are 3 verbal explanations provided for 397 activities (Park et al. 2018). Since this dataset does not have object level annotations, we manually annotated 150 images with 600 object categories.

The classifier for CUB dataset is the compact bilinear pooling model which has an accuracy of 84% (Gao et al. 2016). In FLEX, visual features are embedded to 512-dimensional space whereas words are embedded to 1000-dimensional space. LSTM hidden state dimension is 1000.

For the MPII dataset, we fine-tuned a ResNet-50 pretrained on Imagenet. We learn a 256-dimensional embedding space for words and a 512-dimensional space for visual features. LSTM hidden state dimension is 512.

We use publicly available GVE and MME codes. For CUB dataset, we use the trained GVE model provided by (Hendricks et al. 2016), while the MME model is trained using visual features extracted by the classifier in (Gao et al. 2016). To ensure fair comparison on the MPII dataset, we train both GVE and MME models using visual features extracted from our classification model. In all experiments, explanations are conditioned on the predicted class.

Evaluation Metric

The standard image captioning metrics like BLEU (Papineni et al. 2002), METEOR (Banerjee and Lavie 2005) and CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015) measure how similar a generated explanation is to the ground truth description. However, these metrics do not capture whether the generated justification truly corresponds to the features used by the model to make decision.

Here, we propose a *decision-relevance* measure called DREL to determine how well the generated linguistic description matches the visual features used by a model in its prediction. Given the fine-grained annotations for each image that describe objects in the image, we use the heat maps generated by GradCAM (Selvaraju et al. 2016) to identify objects that are responsible for the model's decision. For CUB, this is the bird's body part that is closest to the maximum point in the heatmap. For MPII, these are objects whose bounding boxes overlap with the regions in the heatmap with values greater than some threshold β.

Let O be the set of objects that are responsible for the model's decision, and OT be the set of words in the ground truth description that refer to the objects in O. Then the metric DREL is defined as

$$DREL = \frac{|OT \cap GJ|}{|OT|} \tag{6}$$

where GJ is the set of words in the generated justification.

Note that DREL requires fine-grained annotations for test images. For CUB dataset, we need part-level annotations (e.g., beak, belly, etc) whereas for MPII dataset, we need object level annotations (e.g., bicycle, ball, etc).

Sensitivity Experiments

We first carry out experiments to set the value of λ in Equation 5. Table 1 shows the DREL results as λ varies. We use $\lambda = 0.1$ for CUB, and $\lambda = 0.001$ for MPII for subsequent experiments as these values give us the best DREL results.

We also examine the effect of various components of FLEX on providing faithful explanations. Table 2 shows the DREL results for variants of FLEX on the CUB and MPII test sets respectively.

The baseline model uses neither the relevance vector nor the weights of CNN features. In other words, it merely describes the image content with no concern for the classifier decision. Thus, it has the least score compared to the other two models.

On the other hand, while "FLEX w/o relevance vector" provides more faithful explanations compared to the baseline model, it does not predict decision relevant words well since it uses only the weighted CNN features. FLEX provides most faithful explanations indicating the effectiveness of using weighted CNN features and relevance vector.

λ	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}	0
CUB	17.39	17.85	17.35	17.11	17.02	16.48
MPII	15.01	15.46	15.19	16.11	15.36	14.13

Table 1: DREL as λ varies on CUB and MPII.

	CUB	MPII
FLEX	**17.85**	**16.11**
FLEX w/o relevance vector	16.48	14.13
Baseline	13.67	13.48

Table 2: DREL Results of FLEX variants.

Comparative Experiments

Next, we compare the performance of FLEX with state-of-art verbal explainers GVE and MME. Table 3 shows the DREL results on the CUB and MPII test sets respectively. We observe that our framework achieved the highest scores compared to GVE and MME in both datasets, indicating that FLEX can provide explanations with more decision-relevant features. In contrast, GVE and MME provide almost the same explanation for all images of the same class.

Table 4 compares the BLEU-4 scores of FLEX, GVE and MME on the CUB and MPII datasets. The results show that FLEX achieves comparable BLEU scores with existing methods. Therefore, we can conclude that FLEX does not trade off sentence fluency for faithfulness in generating verbal explanations and provide better interpretations for deep CNNs compared to existing approaches.

	CUB	MPII
FLEX	**17.85**	**16.11**
GVE (Hendricks et al. 2016)	15.67	13.46
MME (Park et al. 2018)	15.02	13.92

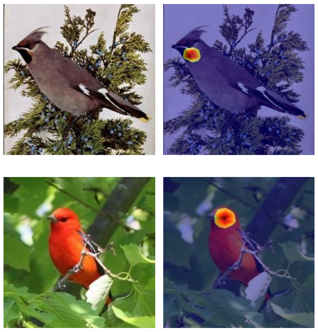
Table 3: DREL Results on CUB and MPII.

	CUB	MPII
FLEX	**30.16**	19.11
GVE (Hendricks et al. 2016)	28.43	13.71
MME (Park et al. 2018)	27.94	**19.88**

Table 4: BLEU-4 Results on CUB and MPII.

Figure 6 and Figure 7 show sample linguistic justifications generated by FLEX, GVE and MME for CUB and MPII datasets respectively. We use the GradCAM visualizations to highlight the regions used by the model for its decisions. We observe that although the explanations generated by GVE and MME describe different visual features in the images, they do not include the objects in the highlighted regions, indicating that these explanations do not reflect how the model makes its decision.

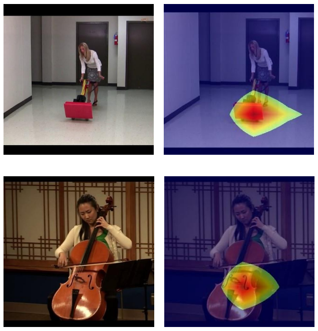
Consider the bird that has been classified as a Bohemian Waxwing in Figure 6. GradCAM shows that the classification model made its decision based on the black throat as highlighted. FLEX is able to generate the linguistic justification that contains *black throat*, which is missing in the explanations generated by GVE and MME.



This is a **Bohemian Waxwing** because
FLEX: This bird has a black crown, a black bill, and a **black throat**.
GVE: This is a gray bird with a black and white wing and a red crown.
MME: This bird has a black crown a white belly and a black bill.

This is a **Scarlet Tanager** because
FLEX: This bird is red in color with a black beak, and **black eye**.
GVE: This is a red bird with black wings and a small beak.
MME: This bird has a red crown and a red breast.

Figure 6: Comparison of justifications generated by FLEX with GVE and MME for CUB dataset.



This is classified to **polishing floors, standing, using electric polishing machine** class because
FLEX: She is **standing** in a room with **a floor polisher** and a rag in her hand.
GVE: She is kneeling on the floor with a carpet and is wearing exercise clothing.
MME: She is holding a mop and is in the middle of moving a mop.

This is classified to **cello, sitting** class because
FLEX: She is **sitting** on a chair and holding **a cello** in her hands.
GVE: She is **sitting** on a chair and playing it with a bow.
MME: She is standing in a room playing a double base.

Figure 7: Comparison of justifications generated by FLEX with GVE and MME for MPII dataset.

Similarly, we see that the explanations generated by FLEX for MPII images in Figure 7 correctly describe poses and equipment used for the different activities. In contrast, GVE and MME seem to have difficulty in providing the correct description for the human activities in these images.

Insights into Incorrect Model Decision

Generating justification that reveals the features used by a model in its decision is also useful when the model gives an incorrect classification. Figure 8 and Figure 9 show the justifications of two images which have been wrongly classified. By comparing these images with other images (on the right) from the predicted class, we observe that justification generated by FLEX describes features which are common across the two classes, that is, the correct class of the image and the predicted class. This common features may have caused the classifier to make the wrong prediction.

For instance, both Fox Sparrow and Sage Thrasher have a short pointy bill and speckled belly and breast. Similarly, both Orange Crowned Warbler and Tennessee Warbler have yellow belly and black wings.

We evaluate the performance of our framework in revealing causes for mis-classifications by calculating the percentage of misclassified images for those FLEX provided correct insights.We consider that a generated explanation provides correct insight for a misclassified image if all the features mentioned in the explanation are common to both the correct class and the predicted class. FLEX provides correct insights for 70.51% of misclassified examples. Such insights may alert users that there may be insufficient training images to help the model to differentiate the two classes.

Generate Explanations for Unseen Classes

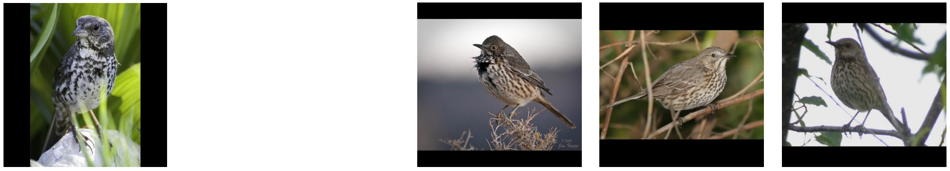
Collecting ground truth descriptions for each class is laborious and expensive. The ability to explain decisions for images of unseen classes where there is no ground truth descriptions is an advantage that FLEX has over GVE and MME. We demonstrate this advantage of FLEX by training the LSTM on CUB dataset leaving out two classes, namely Black Tern and American Goldfinch.

	DREL	BLEU-4
FLEX	**36.72**	**32.70**
GVE (Hendricks et al. 2016)	32.35	27.96
MME (Park et al. 2018)	13.79	26.80

Table 5: DREL and BLEU-4 results on Black Tern and American Goldfinch classes when models are trained without instances from those classes.

Table 5 shows DREL and BLEU-4 results of FLEX, GVE and MME on the left out classes. Since FLEX learns to associate visual features to words and describe decision relevant features, it can provide more faithful and meaningful explanations for CNN decisions of classes not used in training.

FLEX: *This bird has **a speckled belly and breast** with **a short pointy bill**.*

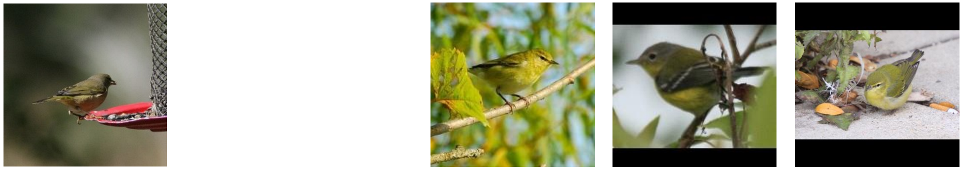


(a) Fox Sparrow

(b) Typical images of Sage Thrasher

Figure 8: Fox Sparrow misclassified as Sage Thrasher.

FLEX: *This bird has **wings that are black** and has **a yellow belly***



(a) Orange Crowned Warbler

(b) Typical images of Tennessee Warbler

Figure 9: Orange Crowned Warbler misclassified as Tennessee Warbler.

(a) Black Tern

FLEX : This bird has **a long black bill**, a white throat, and a white speckled breast.
GVE : This bird has a white belly and breast with a black crown and white bill.
MME : This bird has a white belly a black belly and a black crown.

(b) American Goldfinch

FLEX : This bird has **a yellow belly and breast** and **white wingbars**.
GVE : This bird has **a yellow belly and breast** with a black crown and white crown.
MME : This bird has a yellow head a black breast and a black crown.

Figure 10: FLEX, GVE and MME explanations for unseen classes.

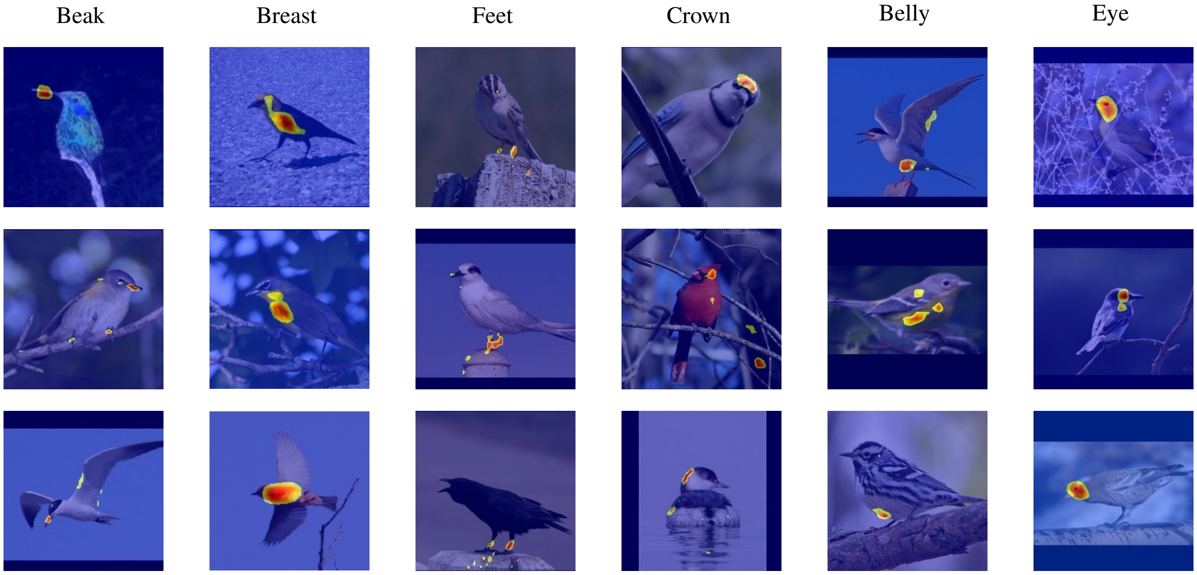


Figure 11: Annotations via a word to feature map association.

Figure 10 shows the explanations generated for sample images from the Black Tern and American Goldfinch classes. FLEX explanations are consistent with the Grad-CAM visualizations, that is, *long black beak* for Figure 10(a) and *yellow belly and breast* and *white wing bars* for Figure 10(b).

Annotate Decision Relevant Features

Another time-consuming task is the fine-grained annotation of objects in images which is useful for semantic segmentation. One key advantage in FLEX is its ability to learn the association between words and feature maps of a CNN. This association allows us to automatically generate annotations for an image.

We extract feature maps from the CNN and calculate the importance of each feature map towards the model decision as in Equation 1. Next, each feature map is multiplied by its importance and values less than a threshold are suppressed to create a heat map. Finally, we resize the heatmap, overlay on the image and annotate it with the word associated with the feature map.

Figure 11 shows the annotations obtained for the CUB dataset. We observe that different parts of the birds such as beak, breast and feet have been correctly annotated. The CUB dataset has 15 part-annotations per image. Our method can correctly annotate at least one part in 88% of images, and can correctly annotate all 15 parts in 35% of images.

Conclusion

In this paper, we have introduced a novel framework that generates linguistic justifications to explain the decisions of a CNN. The proposed framework extracts information about decision relevant features from the model being interpreted, and trains a justification generator using a new objective function to ensure that the generated justifications are relevant to the model decision. Further, we propose a new metric to evaluate how much of the true model reasoning has been revealed in the justifications. Experimental results on the CUB and MPII datasets indicate that the proposed framework outperforms state-of-the-art explanation generators and the justifications generated by our framework can reveal the true rationale behind model decisions.

Acknowledgements

This work is supported by grant R-252-000-A19-490.

References

Andriluka, M.; Pishchulin, L.; Gehler, P.; and Schiele, B. 2014. 2d human pose estimation: New benchmark and state of the art analysis. In *Computer Vision and Pattern Recognition*, 3686–3693.

Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K. R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* 10(7):1–46.

Banerjee, S., and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Smmarization*, 65–72.

Barratt, S. 2017. Interpnet: Neural introspection for interpretable deep learning. In *Interpretable ML Symposium, 31st Conference on Neural Information Processing Systems*.

Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; and Elhadad, N. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *ACM SIGKDD*, 1721–1730.

Gao, Y.; Beijbom, O.; Zhang, N.; and Darrell, T. 2016. Compact bilinear pooling. In *Computer Vision and Pattern Recognition*, 317–326.

Hendricks, L. A.; Akata, Z.; Rohrbach, M.; Donahue, J.; Schiele, B.; and Darrell, T. 2016. Generating visual explanations. In *European Conference on Computer Vision*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Kim, B.; Gilmer, J.; Viegas, F.; Erlingsson, U.; and Wattenberg, M. 2017. Tcav: Relative concept importance testing with linear concept activation vectors. *arXiv:1711.11279*.

Olah, C.; Satyanarayan, A.; Johnson, I.; Carter, S.; Schubert, L.; Ye, K.; and Mordvintsev, A. 2018. The building blocks of interpretability. *Distill* 3(3):e10.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 311 – 318.

Park, D. H.; Hendricks, L. A.; Akata, Z.; Rohrbach, A.; Schiele, B.; Darrell, T.; and Rohrbach, M. 2018. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Computer Vision and Pattern Recognition*.

Reed, S.; Akata, Z.; Lee, H.; and Schiele, B. 2016. Learning deep representations of fine-grained visual descriptions. In *Computer Vision and Pattern Recognition*, 49–58.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *ACM SIGKDD*, 1135–1144.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2016. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *arXiv 1610.02391*.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv:1312.6034*.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2015. Striving for simplicity: The all convolutional net. In *Workshop - ICLR*, 1–14.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *Computer Vision and Pattern Recognition*, 4566–4575.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. Caltech-uscd birds-200-2011 dataset. *California Institute of Technology*.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833.