Rethinking Softmax with Cross-Entropy: Neural Network Classifier as Mutual Information Estimator

Anonymous CVPR submission

Paper ID 4223

Abstract

Mutual information is widely applied to learn latent representations of observations, whilst its implication in classification neural networks remain to be better explained. In this paper, we show that optimising the parameters of classification neural networks with softmax cross-entropy is equivalent to maximising the mutual information between inputs and labels under the balanced data assumption. Through the experiments on synthetic and real datasets, we show that softmax cross-entropy can estimate mutual information approximately. When applied to image classification, this relation helps approximate the point-wise mutual information between an input image and a label without modifying the network structure. In this end, we propose infoCAM, informative class activation map, which highlights regions of the input image that are the most relevant to a given label based on differences in information. The activation map helps localise the target object in an image. Through the experiments on the semi-supervised object localisation task with two real-world datasets, we evaluate the effectiveness of the information-theoretic approach.

1. Introduction

In 2012, AlexNet makes significant progress toward ILSVRC: the ImageNet Large Scale Visual Recognition Challenge, and surpasses a large margin against all the traditional approaches at the time [14, 15]. As a result, such classification neural networks start playing a crucial role in the contemporary machine learning and computer vision community [15]. Apart from their strong categorisation capability, classification neural networks contribute to other tasks, *e.g.* generative adversarial networks use a classification network for producing visually-realistic images [9], and object segmentation networks such as Mask-RCNN uses classification networks for object detection [10].

The softmax function, or softmax in short, is the basic building block of the final layer on classification models. Previous studies interpret softmax as a function that transforms unnormalised values to probabilities since the outputs of softmax sum up to one and are non-negative [4]. Despite the popularity of such interpretation, softmax under this view seems to be an artificial adjustment to enforce the outputs of classification neural networks satisfying probability axioms. This raises a question on an alternative view of softmax being more than a transformation function.

In this paper, we present an information-theoretic interpretation of softmax with cross-entropy. With a variational form of mutual information, we formally prove that optimising model parameter with the softmax cross-entropy is equal to maximising the mutual information between input data and labels via assuming the uniform distribution on labels. The connection provides an alternative view on the classifier as a mutual information estimator. We further propose a probability-corrected version of softmax which relaxes the uniform distribution condition. Based on experiments with a synthetic dataset, we demonstrate the performance of softmax on mutual information estimation.

The connection between classification and information gives a new intuition on interpreting the output of the classifier. As an application, we investigate the image classification problem, especially targeting a class activation map for weakly-supervised object classification tasks. The class activation map aims to find the region which is the most relevant to the target class. We propose a new approach, dubbed as infoCAM, to compute a class activation map based on the point-wise mutual information obtained by classification. Through the experiments, we evaluate the effectiveness of infoCAM on weakly-supervised object localisation with Tiny-ImageNet [1] and CUB-200-2011 [23] datasets.

In summary, we outline our contributions with the corresponding section as follows:

- In section 3, we prove that classification neural networks that optimise their weights to minimise the softmax cross-entropy are equivalent to the ones that maximise mutual information between inputs and labels with the balanced dataset.

- In section 4, we empirically evaluate the effectiveness of classification mutual information estimator via synthetic and real-world datasets.
- In section 5, we propose infoCAM. A map that reveals the most relevant regions of an image with respect to the target label based on the difference of mutual information.
- In section 6, we demonstrate the performance of the infoCAM on WSOL results, achieving a new state-of-the-art on Tiny-ImageNet.

2. Preliminaries

In this section, we first define the notations used throughout this paper. We then introduce the definition of mutual information and variational forms of mutual information.

2.1. Notation

We let training data consisting of M classes and N labelled instances as $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $y_i \in \mathcal{Y} = \{1, ..., M\}$ is a class label of the input $\mathbf{x}_i$. We let $n_\phi(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^M$ be a neural network parameterised by ϕ, where $\mathcal{X}$ is a space of input $\mathbf{x}$. Without additional clarification, we assume $\mathcal{X}$ to be a compact subset of D-dimensional Euclidean space. We denote by P_{XY} some joint distribution over $\mathcal{X} \times \mathcal{Y}$, with $(\mathbf{X}, Y) \sim P_{XY}$ a pair of random variables. P_X and P_Y are the marginal distributions of $\mathbf{X}$ and Y, respectively. We remove a subscript from the distribution if it is clear from context.

2.2. Variational Bounds of Mutual Information

Mutual information evaluates the mutual dependence between two random variables. The mutual information between $\mathbf{X}$ and Y can be expressed as:

$$\mathbb{I}(\mathbf{X}, Y) = \int_{\mathbf{x}\in\mathcal{X}} \left[\sum_{y\in\mathcal{Y}} P(\mathbf{x}, y) \log \left(\frac{P(\mathbf{x}, y)}{P(\mathbf{x})P(y)} \right) \right] d\mathbf{x}. \tag{1}$$

Equivalently, following [21] we may express the definition of mutual information in Equation 1 as:

$$\mathbb{I}(\mathbf{X}, Y) = \mathbb{E}_{(\mathbf{X},Y)} \left[\log \frac{P(y|\mathbf{x})}{P(y)} \right], \tag{2}$$

where $\mathbb{E}_{(\mathbf{X},Y)}$ is the abbreviations of $\mathbb{E}_{(\mathbf{X},Y)\sim P_{XY}}$. Computing mutual information directly from the definition is, in general, intractable due to integration.

Variational form: Barber and Agakov introduce a common used lower bound of mutual information via a variational distribution Q [3], derived as:

$$\begin{aligned}
\mathbb{I}(\mathbf{X}, Y) &= \mathbb{E}_{(\mathbf{X},Y)} \left[\log \frac{P(y|\mathbf{x})}{P(y)} \right] \\
&= \mathbb{E}_{(\mathbf{X},Y)} \left[\log \frac{Q(y|\mathbf{x})}{P(y)} \frac{P(y|\mathbf{x})}{Q(y|\mathbf{x})} \right] \\
&= \mathbb{E}_{(\mathbf{X},Y)} \left[\log \frac{Q(y|\mathbf{x})}{P(y)} \right] + \underbrace{\mathbb{E}_{(\mathbf{X},Y)} \left[\log \frac{P(y|\mathbf{x})}{Q(y|\mathbf{x})} \right]}_{D_{KL}(P(\mathbf{x},y)||Q(\mathbf{x},y))} \\
&\geq \mathbb{E}_{(\mathbf{X},Y)} \left[\log \frac{Q(y|\mathbf{x})}{P(y)} \right].
\end{aligned} \tag{3}$$

The inequality in Equation 3 holds since KL divergence maintains non-negativity. This lower bound is tight when variational distribution $Q(y|\mathbf{x})$ converges to posterior distribution $P(y|\mathbf{x})$, i.e., $Q(y|\mathbf{x}) = P(y|\mathbf{x})$.

The form in Equation 3 is, however, still hard to compute since it is not easy to make a tractable and flexible variational distribution $Q(y|\mathbf{x})$. Variational distribution $Q(y|\mathbf{x})$ can be considered as a constrained function which has to satisfy the probability axioms. Especially, the constrain is challenging to model with a function estimator such as a neural network. To relax the function constraint, McAllester *et al.* [18] further apply reparameterisation and define $Q(y|\mathbf{x})$ in terms of an unconstrained function f_ϕ parameterised by ϕ as:

$$Q(y|\mathbf{x}) = \frac{P(y)}{E_{y'\sim P_Y}[\exp(f_\phi(\mathbf{x}, y'))]} \exp(f_\phi(\mathbf{x}, y)). \tag{4}$$

As a consequence, the variational lower bound of mutual information $\mathbb{I}(\mathbf{X}, Y)$ can be rewritten with function f_ϕ as:

$$\mathbb{I}(\mathbf{X}, Y) \geq \mathbb{E}_{(\mathbf{X},Y)} \left[\log \frac{\exp(f_\phi(\mathbf{x}, y))}{E_{y'}[\exp(f_\phi(\mathbf{x}, y'))]} \right]. \tag{5}$$

One can estimate mutual information without any constraint on f. Through the reparameterisation, the MI estimation can be cast as an optimisation problem.

3. Connecting Mutual Information to Softmax

In this section, we show the connection between mutual information and the classification neural network.

3.1. Softmax with Balanced Dataset

Softmax is widely used to map an output of neural network into a categorical probabilistic distribution for classification. Given neural network $n(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^M$, softmax $\sigma : \mathbb{R}^M \to \mathbb{R}^M$ is defined as:

$$\sigma(n(\mathbf{x}))_y = \frac{\exp(n(\mathbf{x})_y)}{\sum_{y'=1}^M \exp(n(\mathbf{x})_{y'})}. \tag{6}$$

Expected cross-entropy is often employed to train a neural network with softmax output. The expected cross-entropy loss is

$$L = -\mathbb{E}_{(\mathbf{x},Y)}[n(\mathbf{x})_y - \log(\sum_{y'=1}^{M} \exp(n(\mathbf{x})_{y'}))], \quad (7)$$

where the expectation is taken over the joint distribution of X and Y. Given a training set, one can train the model with an empirical distribution of the joint distribution. We present an interesting connection between cross-entropy with softmax and mutual information in the following theorem.

Theorem 1. *Let $f_\phi(\mathbf{x}, y)$ be $\sigma(n(\mathbf{x}))_y$. The lower bound of mutual information in Equation 5 can be obtained by minimising the expected cross-entropy with softmax for classification up to constant $\log M$ under the uniform label distribution.*

Proof. Let $f_\phi(\mathbf{x}, y) = \sigma(n(\mathbf{x}))_y$, then the lower bound is

$$\mathbb{E}_{(\mathbf{x},Y)}\left[\log \frac{\exp(n(\mathbf{x})_y)}{E_{y'}[\exp(n(\mathbf{x})_{y'})]}\right]. \quad (8)$$

If the distribution of the label is uniform then, it can be rewritten as

$$\mathbb{E}_{(\mathbf{x},Y)}\left[\log \frac{\exp(n(\mathbf{x})_y)}{1/M\sum_{y'=1}^{M}\exp(n(\mathbf{x})_{y'})}\right]$$
$$= \mathbb{E}_{(\mathbf{x},Y)}\left[\log \frac{\exp(n(\mathbf{x})_y)}{\sum_{y'=1}^{M}\exp(n(\mathbf{x})_{y'})}\right] + \log M, \quad (9)$$

which is equivalent to the negative expected cross-entropy loss (7) up to constant $\log M$. Hence, by minimising the cross-entropy, we can obtain the lower bound of mutual information. □

Note that the constant does not change the gradient of the objective. Consequently, the solutions of both the mutual information maximisation and softmax cross-entropy minimisation optimisation problems are the same.

3.2. Softmax with Imbalanced Dataset

The uniform label distribution assumption in Theorem 1 is restrictive since we cannot access to the true label distribution, often assumed to be non-uniform. To relax the restriction, we propose a probability-corrected softmax (PC-softmax):

$$\sigma_p(n(\mathbf{x}))_y = \frac{\exp(n(\mathbf{x})_y)}{\sum_{y'=1}^{M} P(y')\exp(n(\mathbf{x})_{y'})}, \quad (10)$$

where $P(y')$ is a distribution over label y'. Instead we can optimise the revised softmax with empirical distribution $\widehat{P}(y')$ estimated from training set. We show the equivalence between optimising the classifier and maximising mutual information with the new softmax below.

y	μ	# samples	$p(y)$
0	$\mathbf{0}$	6,000	0.07
1	$+\mathbf{2}$	12,000	0.13
2	$-\mathbf{2}$	18,000	0.20
3	$+\mathbf{4}$	24,000	0.27
4	$-\mathbf{4}$	30,000	0.33

Table 1: Synthetic dataset description. μ is a mean vector for each Gaussian distribution. # samples denotes the number (resp. prior distribution) of samples with the non-uniform prior assumption. For the test with the uniform prior assumption, we use 12,000 samples from each distribution.

Theorem 2. *The mutual information between two random variable X and Y can be obtained via the infimum of cross-entropy with PC-softmax in Equation 10 under a mild condition on n.*

Proof. First, it can be easily shown that we can relax the uniform assumption with PC-softmax. We then show that the class of functions modelled by $n : \mathcal{X} \to \mathbb{R}^M$ is the same as those of $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Y is a categorical variable. Hence, unconstrained function f can be decomposed into a set of functions indexed by y, i.e., $f = \{f_y\}_{y=1}^{M}$. Under a mild condition, $n(x)_y$ can approximate any continuous function by the universal approximation theorem [13]. We conclude the proof by letting $n(x)_y$ be f_y. □

Mutual information is often used in generative models to find the maximally informative representation of observation [12, 25], whereas its implication in classification has been unclear so far. The results of this section imply the classification neural network with softmax optimises its weights to maximise the mutual information between inputs and labels under the uniform label assumption. We further study an application of this implication in section 5 to tackle the weakly supervised object localisation task.

4. Estimating MI via Classification

In the previous section, we show that classification neural networks can be utilised to measure the mutual information (MI) between continuous and discrete distributions. We measure the empirical performance of softmax based mutual information estimator via synthetic datasets.

To construct a synthetic data with a pair of continuous and discrete variables, we employ a Gaussian mixture

Dim.	Acc.(%)	Mutual information MC	Mutual information softmax
1	74	1.03	0.99
2	85	1.30	1.28
5	94	1.54	1.48
10	98	1.60	1.54

(a) Results with balanced datasets.

Dim.	Acc.(%)	Mutual information MC	softmax	PC-softmax
1	79 / 79	1.02	1.11	0.96
2	87 / 88	1.23	1.31	1.20
5	93 / 95	1.44	1.41	1.31
10	95 / 96	1.48	1.36	1.34

(b) Results with imbalanced datasets. Acc. stands for the classification accuracy with Softmax and PC-Softmax, respectively.

Table 2: Mutual information estimation results with softmax-based classification neural networks. Dim. means input data dimension. Acc. stands for the classification accuracy. MC represents the estimated mutual information via Monte Carlo methods.

model:

$$P(x) = \sum_{y=1}^{M} P(y)\mathcal{N}(\mathbf{x}|\mu_y, \mathbf{\Sigma}_y)$$
$$P(x|y) = \mathcal{N}(\mathbf{x}|\mu_y, \mathbf{\Sigma}_y),$$

where $P(y)$ is a prior distribution over the labels.

For the experiments, we use five mixtures of isotropic Gaussian, each of which has a unit diagonal covariance matrix with different means. We set the parameters of the mixtures to make them overlap the significant proportion of their distributions.

We generate two sets of datasets: one with the uniform prior and the other with the non-uniform prior distribution over labels, $p(y)$. For the uniform prior, we sample 12,000 data points from each Gaussian, and for the non-uniform prior, we sample unequal number of data points from each Gaussian. In addition, we increase the dimensionality of Gaussian distribution from 1 to 10. The detailed statistics for the Gaussian parameters and the number of samples are available at Table 1. To train classification models, we divide the dataset into training, validation and test. We use the validation set to find the best parameter configuration of the classifier.

We aim to compare the difference of true and softmax-based estimated mutual information $\mathbb{I}(\mathbf{X}, Y)$. The true mutual information defined as Equation 1 is, however, intractable. We thus approximate it via Monte Carlo (MC) methods using the true probability density function, expressed as:

$$\mathbb{I}(\mathbf{X}, Y) \approx \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{P(\mathbf{x}_i|y_i)}{P(\mathbf{x}_i)}\right), \tag{11}$$

where $(\mathbf{x}_i, y_i)$ forms a paired sample. Equation 11 attains equality as N approaches infinity. To estimate mutual information via classification, we train classification models with two versions of softmax.

Once we choose the best model based on validation set, we estimate mutual information $\mathbb{I}(\mathbf{X}, Y)$ with a version of softmax to train the model via Equation 9 or Equation 10. We use four layers of a feed-forward neural network with the ReLU as an activation for internal layers and softmax as an output layer[1]. We measure the performance of two softmax versions on classification and mutual information estimation tasks.

Table 2a summarises the experimental results with the balanced dataset. With the balanced dataset, there is no difference between softmax and PC-softmax. Note that the MC estimator has an access to explicit model parameters for estimating mutual information, whereas the softmax estimator measures mutual information based on the model outputs without accessing to the true distribution. We could not find a significant difference between MC and softmax estimator. Table 2b summarises the experimental results with the imbalanced dataset. The results show that the PC-softmax slightly under-estimates mutual information to compare with the other two approaches. It is worth noting that the classification accuracy of PC-softmax consistently outperforms the original softmax.

We further test the classification performance of softmax and PC-softmax with two real-world datasets: MNIST and CUB-200-2011 [23]. To MNIST, we use a subset of the original dataset such that instance numbers for every class are all the same in order to construct balanced MNIST, while randomly subsample one half of instances for 1/2 classes to make imbalanced MNIST. That is, randomly drop one half of images for digit 0, 2, 4, 6 and 8. To CUB-200-2011, we follow the same training and validation splits as in [7] in order to compare with their results. As a result of such splitting, the training set is approximately balanced, where out of the total 200 classes, 196 of them contain 30 instances and the rest 6 classes include 29 instances. To construct an imbalanced dataset, similar to MNIST, we randomly drop one half of instances from 1/2 bird classes.

We adopt a simple convolutional neural network as a classifier for MNIST. The model contains two convolutional layers, each followed by a max pooling layer and the ReLU

[1] All model details used in this paper are available in the supplementary material.

Dataset	MNIST		CUB-200-2011	
	Bal.	Imbal.	Bal.	Imbal.
softmax	97.95	96.81	89.23	89.21
PC-softmax	97.91	96.86	89.18	89.73*

(a) Classification accuracy (%).

Dataset	MNIST		CUB-200-2011	
	Bal.	Imbal.	Bal.	Imbal.
softmax	97.95	95.05	89.21	84.63
PC-softmax	97.91	96.30*	89.16	87.69*

(b) Average per-class accuracy (%).

Table 3: Classification accuracy of using softmax and PC-softmax. Numbers of instances for different labels are the same within a balanced dataset and are significantly distinct within an imbalanced dataset. Values with asterisk denote p-values less than 0.05 with the Mann-Whitney U test.

activation, followed by two fully connected layers with the final softmax. For CUB-200-2011, we apply the same architecture as Inception-V3, which demonstrates the state-of-the-art classification performance in CUB-200-2011 after being fine-tuned [7]. We measure both the micro accuracy and the average per-class accuracy of the two softmax versions on both datasets, where the latter alleviates the dominance of the majority classes in unbalanced datasets. The classification results are shown in Table 3. PC-softmax is significantly more accurate than softmax on imbalanced datasets in terms of the average per-class accuracy.

5. Weakly Supervised Object Localisation

In the following section, we show how implication of the previous section can be applied to tackle real-world problems. We first introduce the concept and definition of the class activation map, and show how to apply it to the weakly supervised object localisation (WSOL) task. We then propose the Informative Class Activation Map (infoCAM) based on the connection between mutual information and softmax.

5.1. CAM: Class Activation Map

Contemporary classification CNNs such as AlexNet [14] and Inception [22] consists of stacks of convolutional layers interleaving with pooling layers for extracting visual features. These convolutional layers result in feature maps, which is a collection of 2-dimensional grids. The size of the feature map depends on the structure of convolution and pooling layers. Often the feature map is smaller than the original image. The number of a feature map corresponds to the number of convolution filters. The feature maps from the final convolutional layer are usually averaged, flattened and fed into the fully-connected layer for classification [17]. Given K feature maps $g_1, .., g_K$, the fully-connected layer consists of weight matrix $W \in \mathbb{R}^{M\times K}$, where w_k^y represents the scalar weight corresponding to class y for feature k. We use $g_k(a,b)$ to denote a value of 2-dimensional spatial point (a,b) with feature k in map g_k. In [6], the authors propose a way to interpret the importance of each point in feature maps. The importance of spatial point (a,b) for class y is defined as a weighted sum over features:

$$M_y(a,b) = \sum_k w_k^y g_k(a,b). \tag{12}$$

We redefine $M_y(a,b)$ as an intensity of the point (a,b). The collection of these intensity over all grid points forms a class activation map (CAM), which highlights the most relevant region the in feature space for classifying y. The input going to the softmax layer corresponding to the class label y is:

$$\sum_{a,b} M_y(a,b) = n(\mathbf{x})_y. \tag{13}$$

Intuitively, weight w_k^y indicates the overall importance of the kth feature to class y, and intensity $M_y(a,b)$ implies the importance of the feature map at spatial location (a,b) leading to the classification of image $\mathbf{x}$ to y.

The aim of WSOL is to identify the region containing the target object in an image given a label without having a pixel-level supervision. Previous work tackles WSOL task by creating a bounding box from the CAM [6]. They create a bounding box within a CAM. Such CAM contains all important locations that exceed a certain intensity threshold. The box is then upsampled to match the size of the original image.

5.2. InfoCAM: Informative Class Activation Map

In section 3, we show that softmax classifier carries an explicit implication between inputs and labels in terms of information theory. We extend the notion of mutual information from a pair of an input image and a label to regions of the input image and the label to capture the regions that have high mutual information with labels.

To reduce clutter, we assume that there is only one feature map, *i.e.* $K = 1$. However, the following results can be easily applied to the general cases where $K > 1$ without loss of generality. We then introduce a region R containing a subset of grid points in feature map g.

Mutual information is an expectation of the pointwise mutual information (PMI) between two variables, *i.e.* $\mathbb{I}(\mathbf{X}, Y) = \mathbb{E}[\text{PMI}(\mathbf{x}, y)]$. Given two instances of variables, we can estimate their PMI via Equation 9, *i.e.*

$$\text{PMI}(\mathbf{x}, y) = n(\mathbf{x})_y - \log \sum_{y'=1}^{M} \exp(n(\mathbf{x})_{y'}) + \log M,$$

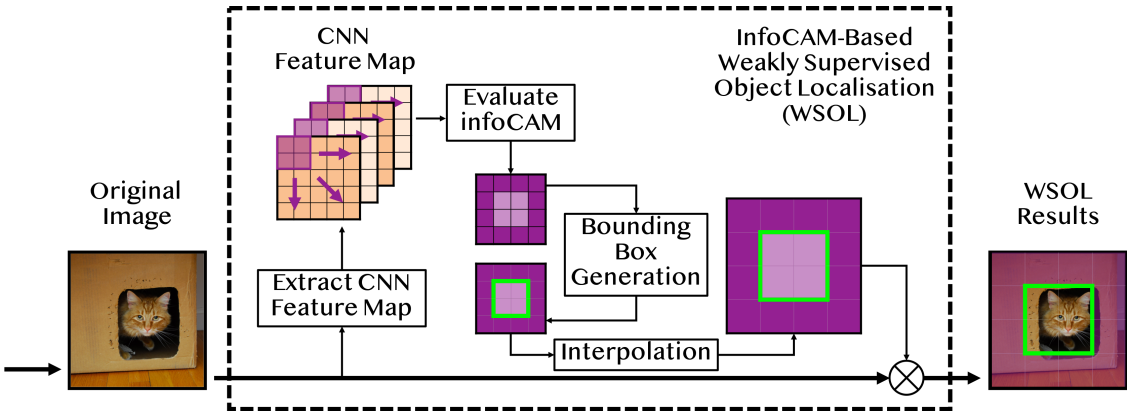

Figure 1: A visualisation of the infoCAM procedure for the WSOL task. The task aims to draw a bounding box for the target object in the original image. The procedure includes: 1) feed input image into a CNN to extract its feature maps, 2) evaluate PMI difference between the true and the other labels of input image for each region within feature maps, 3) generate the bounding box by keeping the regions exceeding certain infoCAM values and find the largest connected region and 4) interpolate and map the bounding box to the original image.

The PMI is close to $\log M$ if y is the maximum argument in log-sum-exp. To find a region which is the most beneficial to the classification, we compute the difference between PMI with true label and the average of the other labels and decompose it into a point-wise summation as

$$\begin{aligned}\text{Diff}(\text{PMI}(\mathbf{x})) &= \text{PMI}(\mathbf{x}, y^*) - \frac{1}{M-1}\sum_{y'\neq y^*}\text{PMI}(\mathbf{x}, y'),\\ &= \sum_{(a,b)\in g} w^{y*}g(a,b) - \frac{1}{M-1}\sum_{y'\neq y^*} w^{y'}g(a,b).\end{aligned}$$

The point-wise decomposition suggests that we can compute the PMI differences with respect to a certain region. Based on this observation, we propose a new CAM, named informative CAM or infoCAM, with the new intensity function $M_y^{\text{Diff}}(R)$ between region R and label y defined as follows:

$$M_y^{\text{Diff}}(R) = \sum_{(a,b)\in R} w^y g(a,b) - \frac{1}{M-1}\sum_{y'\neq y} w^{y'}g(a,b). \tag{14}$$

The infoCAM highlights the region which decides the classification boundary against the other labels. The region based metric smooths the importance of a certain point across the region. Moreover, we further simplify Equation 14 to be the difference between PMI with true and the most-unlikely labels according to the classifier's outputs, denoting as infoCAM+, with the new intensity:

$$M_y^{\text{Diff}^+}(R) = \sum_{(a,b)\in R} w^y g(a,b) - w^{y'}g(a,b), \tag{15}$$

where $y' = \arg\min_m \sum_{(a,b)\in R} w^m g(a,b)$.

The complete procedure of WSOL with infoCAM is visually illustrated in Figure 1. We first feed input image into a CNN to extract its feature maps. Then instead of computing CAM of the feature map, we compute infoCAM of varying regions from the input image and the class label. Afterwards, we generate the bounding box for the object by preserving regions surpassing a certain intensity level. Then we generate the bounding box that covers the largest connected remaining regions [26]. Finally, we interpolate the generated bounding box to the original image size and merge the two.

6. Localising Object with infoCAM

In this section, we demonstrate experimental results with infoCAM on WSOL. We first describe the experimental settings and then present the results.

6.1. Experimental settings

We evaluate WSOL performance on CUB-200-2011 [23] and Tiny-ImageNet [1]. CUB-200-2011 consists of 200 bird specifies, including 5,994 training and 5,794 validation images. Each bird class contains relatively the same number of instances, thus the dataset is approximately balanced. Since the dataset only depicts birds, not including other kinds of objects, varieties due to class difference are subtle [8]. Therefore, CNN-based classifiers turn to concentrate on the most discriminative areas within an image whilst disregarding other regions that are similar among all the birds [24]. Such nuance-only detection can lead to localisation accuracy degradation [6].

Tiny-ImageNet is a reduced version of ImageNet in terms of both class number, number of instances per class and image resolution. It includes 200 classes, and each

		GT Loc. (%)	Top-1 Loc. (%)
VGG	CAM	42.49	31.38
	CAM (ADL)	71.59	53.01
	infoCAM	52.96	39.79
	infoCAM (ADL)	73.35	53.80
	infoCAM+	59.43	44.40
	infoCAM+ (ADL)	**75.89**	54.35
ResNet	CAM	61.66	50.84
	CAM (ADL)	57.83	46.56
	infoCAM	64.78	53.22
	infoCAM (ADL)	67.75	54.71
	infoCAM+	68.99	**55.83**
	infoCAM+ (ADL)	69.63	55.20

(a) Localisation results on CUB-200-2011.

		GT Loc. (%)	Top-1 Loc. (%)
VGG	CAM	53.49	33.48
	CAM (ADL)	52.75	32.26
	infoCAM	55.50	34.27
	infoCAM (ADL)	53.95	33.05
	infoCAM+	55.25	34.27
	infoCAM+ (ADL)	53.91	32.94
ResNet	CAM	54.56	40.55
	CAM (ADL)	52.66	36.88
	infoCAM	**57.79**	**43.34**
	infoCAM (ADL)	54.18	37.79
	infoCAM+	57.71	43.07
	infoCAM+ (ADL)	53.70	37.71

(b) Localisation results on Tiny-ImageNet.

Table 4: Localisation results of CAM and infoCAM on CUB-2011-200 and Tiny-ImageNet. InfoCAM outperforms CAM on localisation of objects with the same model architecture. Bold values represent the highest accuracy for a certain metric.

consists of 500 training and 50 validation images, thus is balanced. Unlike CUB-200-2011 comprising only birds, Tiny-ImageNet contains a wide range of objects from animals to daily supplies. Compared with the full ImageNet, training classifiers on TinyImageNet is faster due to image resolution reduction and quantity shrink, yet classification becomes more challenging [19].

To perform an evaluation on localisation, we first need to generate a bounding box for the object within an image. We generate a bounding box in the same way as in [26]. Specifically, after evaluating infoCAM within each region of an image, we only reserve the regions whose infoCAM values are more than 20% of the maximum infoCAM and

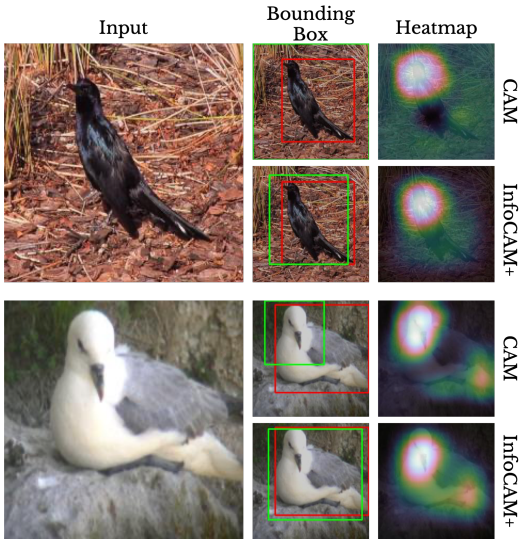

Figure 2: Visualisation of comparison between CAM and infoCAM+. Red and green boxes represent the ground truth and prediction, respectively. Brighter regions represent higher CAM or infoCAM+ values.

abandon all the other regions. Then, we draw the smallest bounding box that covers the largest connected component.

We follow the same evaluation metrics in [6] to evaluate localisation performance with two accuracy measures: 1) localisation accuracy with known ground truth class (GT Loc.), 2) top-1 localisation accuracy (Top-1 Loc.). GT Loc. draws the bounding box from the ground truth of image labels, whereas Top-1 Loc. draws the bounding box from the most likely image label and also requires correct classification. The localisation of an image is judged as correct when the intersection over union of the estimated bounding box and the ground-truth bounding box is greater than 50%.

We adopt the same network architectures and hyperparameters as in [6], which shows the current state-of-the-art performance. Specifically, the network backbones can be ResNet50 [11] and a variation of VGG16 [22], in which the fully connected layers are replaced with global average pooling (GAP) layers to reduce the number of parameters. The traditional softmax is used as the final layer since both datasets are well balanced. InfoCAM requires the region parameter R. We apply a square region for the region parameter R. The size of the region R is set as 5 and 4 for VGG and ResNet in CUB-200-2011, respectively, and 3 for both VGG and ResNet in Tiny-ImageNet.

These models are tested with the Attention-based Dropout Layer (ADL) to tackle the localisation degradation problem [6]. ADL is designed to randomly abandon some of the most discriminative image regions during training to ensure CNN-based classifiers cover the entire object. The ADL-based approaches demonstrate state-of-the-art perfor-

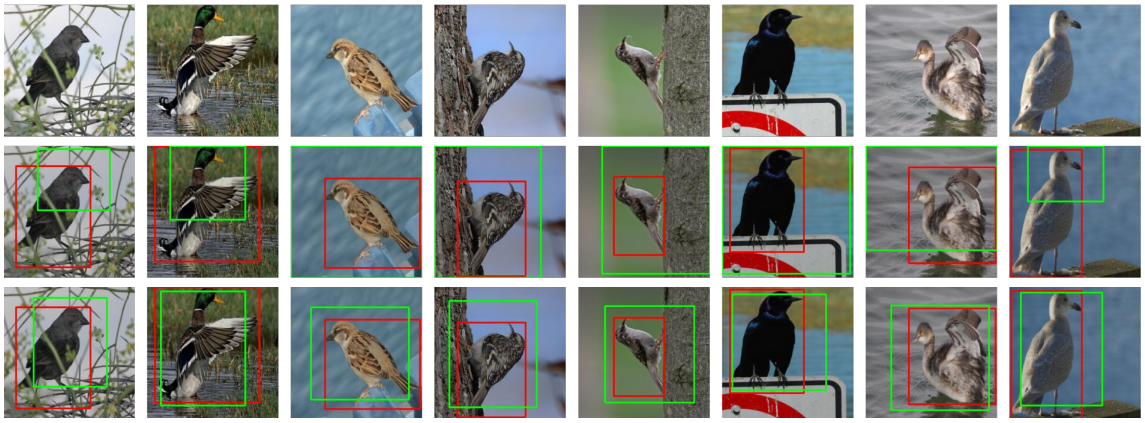

Figure 3: Visualisation of localisation with ResNet50 without using ADL on CUB-200-2011. Images in the second and the third row correspond to CAM and infoCAM+, respectively. Estimated and ground-truth bounding boxes are green and red, separately.

mance in CUB-200-2011 [6] and Tiny-ImageNet [5] for the WSOL task as far as we know and are computationally efficient. We test ADL with infoCAMs to enhance WSOL capability.

To prevent overfitting in the test dataset, we evenly split the original validation images to two data piles, one still for validation during training and the other acting as the final test dataset. We pick the trained model from the epoch that demonstrates the highest top-1 classification accuracy in the validation dataset and report the experimental results with the test dataset. All experiments are running on two Nvidia 2080-Ti GPUs, with PyTorch deep learning framework [20].

6.2. Experimental Results

Table 4 shows the localisation results on CUB-200-2011 and Tiny-ImageNet. The results demonstrate that infoCAM can consistently improve accuracy than the original CAM for WSOL under a wide range of networks and datasets. Both infoCAM and infoCAM+ perform comparable to each other. ADL improves the performance of both models with CUB-200-2011 datasets, but it worsens the performance with Tiny-ImageNet. We conjecture that dropping any part of a Tiny-ImageNet image with ADL significantly influences classification since the images are relatively small.

Figure 2 highlights the difference between CAM an infoCAM. The figure suggests that infoCAM gives relatively high intensity on the object to compare with that of CAM, which only focuses on the head part of the bird. Figure 4 in Appendix presents the additional examples of visualisation for comparing localisation performance of CAM to infoCAM, both without the assistance of ADL[2]. From these visualisations, we notice the bounding boxes generated from infoCAM tight closer to the objects than the original CAM. That is, infoCAM turns to precisely cover the areas where objects existing, no extraneous or lacking. For example, CAM highlights bird heads in CUB-200-2011, whereas infoCAM covers bird bodies as well.

Ablation Study: InfoCAM differs from CAM in two ways: 1) the new intensity function and 2) region-based intensity smoothing with parameter R. We conduct ablation study to figure out which feature helps localise objects. The results suggest that both components are indispensable to improve the performance of the localisation. For the detailed results, please refer to Table 6 in Appendix.

7. Conclusion

We have shown the connection between mutual information and softmax classifier through the variational form of mutual information. The connection explains the rational behind softmax cross-entropy from information theoretic perspective, which brings a new insight to understand the classifiers. There exists previous work that names the negative log-likelihood (NLL) loss as maximum mutual information estimation [2, 16]. Despite naming similarity, they do not show the relationship between softmax and mutual information.

We utilise the connection between classification and mutual information to improve the weakly-supervised object localisation task. To this end, we propose a new way to compute the classification activation map, which is based on the difference between PMIs. The experimental results show the practicality of the information theoretic approach. We believe that this opens new ways to understand and interpret how the neural network classifiers work.

[2]Please refer to the supplementary material for more complete CUB-200-2011 and Tiny-ImageNet visualisation results.

References

[1] Tiny imagenet visual recognition challenge. https://tiny-imagenet.herokuapp.com/. Accessed: 2019-11-03. 1, 6

[2] Lalit R Bahl, Peter F Brown, Peter V De Souza, and Robert L Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *proc. icassp*, volume 86, pages 49–52, 1986. 8

[3] David Barber and Felix V Agakov. The im algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*, page None, 2003. 2

[4] John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990. 1

[5] Junsuk Choe, Joo Hyun Park, and Hyunjung Shim. Improved techniques for weakly-supervised object localization. *arXiv preprint arXiv:1802.07888*, 2018. 8

[6] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2219–2228, 2019. 5, 6, 7, 8

[7] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4109–4118, 2018. 4, 5

[8] Abhimanyu Dubey, Otkrist Gupta, Pei Guo, Ramesh Raskar, Ryan Farrell, and Nikhil Naik. Pairwise confusion for fine-grained visual classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 70–86, 2018. 6

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[12] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representation*, 2019. 3

[13] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 3

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 5

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. 1

[16] Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. In *Predicting structured data*. MIT Press, 2006. 8

[17] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *International Conference on Learning Representation*, 2014. 5

[18] David McAllester and Karl Statos. Formal limitations on the measurement of mutual information. *arXiv preprint arXiv:1811.04251*, 2018. 2

[19] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017. 7

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Proceedings of Neural Information Processing Systems*, 2017. 8

[21] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, 2019. 2

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 5, 7

[23] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1, 4, 6

[24] Hongjun Wang, Guangrun Wang, Guanbin Li, and Liang Lin. Camdrop: A new explanation of dropout and a guided regularization method for deep neural networks. In *International Conference on Information and Knowledge Management (CIKM)*, pages 2219–2228, 2019. 6

[25] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017. 3

[26] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 6, 7

		GT Loc. (%)	Top-1 Loc. (%)	Top-1 Class (%)	Top-5 Class (%)
VGG-16-GAP	CAM	42.49	31.38	73.97	91.83
	CAM (ADL)	71.59	53.01	71.05	90.20
	infoCAM	52.96	39.79	-	-
	infoCAM (ADL)	73.35	53.80	-	-
	infoCAM+	59.43	44.40	-	-
	infoCAM+ (ADL)	**75.89**	54.35	-	-
ResNet-50	CAM	61.66	50.84	80.54	94.09
	CAM (ADL)	57.83	46.56	79.22	94.02
	infoCAM	64.78	53.22	-	-
	infoCAM (ADL)	67.75	54.71	-	-
	infoCAM+	68.99	**55.83**	-	-
	infoCAM+ (ADL)	69.63	55.20	-	-

(a) Localisation and classification results on CUB-200-2011.

		GT Loc. (%)	Top-1 Loc. (%)	Top-1 Class (%)	Top-5 Class (%)
VGG-16-GAP	CAM	53.49	33.48	55.25	79.19
	CAM (ADL)	52.75	32.26	52.48	78.75
	infoCAM	55.50	34.27	-	-
	infoCAM (ADL)	53.95	33.05	-	-
	infoCAM+	55.25	34.27	-	-
	infoCAM+ (ADL)	53.91	32.94	-	-
ResNet-50	CAM	54.56	40.55	66.45	86.22
	CAM (ADL)	52.66	36.88	63.21	83.47
	infoCAM	**57.79**	**43.34**	-	-
	infoCAM (ADL)	54.18	37.79	-	-
	infoCAM+	57.71	43.07	-	-
	infoCAM+ (ADL)	53.70	37.71	-	-

(b) Localisation and classification results on Tiny-ImageNet.

Table 5: Evaluation results of CAM and infoCAM on CUB-2011-200 and Tiny-ImageNet. Note that the classification accuracy of infoCAM is the same as those of CAM. InfoCAM always outperforms CAM on localisation of objects under the same model architecture.

A. Further Result

A.1. Localisation and Classification Result

Table 5 is a reproduction of main result with the classification results.

A.2. Ablation Study

A.3. Results of Both Datasets

ADL	R	Subtraction Term	GT Loc. (%)	Top-1 Loc. (%)
N	N	N	42.49	31.38
	N	Y	47.59 ↑	35.01 ↑
	Y	N	53.40 ↑	40.19 ↑
Y	N	N	71.59	53.01
	N	Y	75.78 ↑	54.28 ↑
	Y	N	73.56 ↑	53.94 ↑

(a) Localisation results on CUB-200-2011 with VGG-GAP.

ADL	R	Subtraction Term	GT Loc. (%)	Top-1 Loc. (%)
N	N	N	54.56	40.55
	N	Y	54.29 ↓	40.51 ↓
	Y	N	57.73 ↑	43.34 ↑
Y	N	N	52.66	36.88
	N	Y	52.52 ↓	37.08 ↑
	Y	N	54.15 ↑	37.76 ↑

(b) Localisation results on CUB-200-2011 with ResNet50.

Table 6: Ablation study results on the importance of the region parameter R and the subtraction term within the formulation of infoCAM. Y and N indicates the use of corresponding feature. Arrows indicates the relative performance against the case where both features are not used.

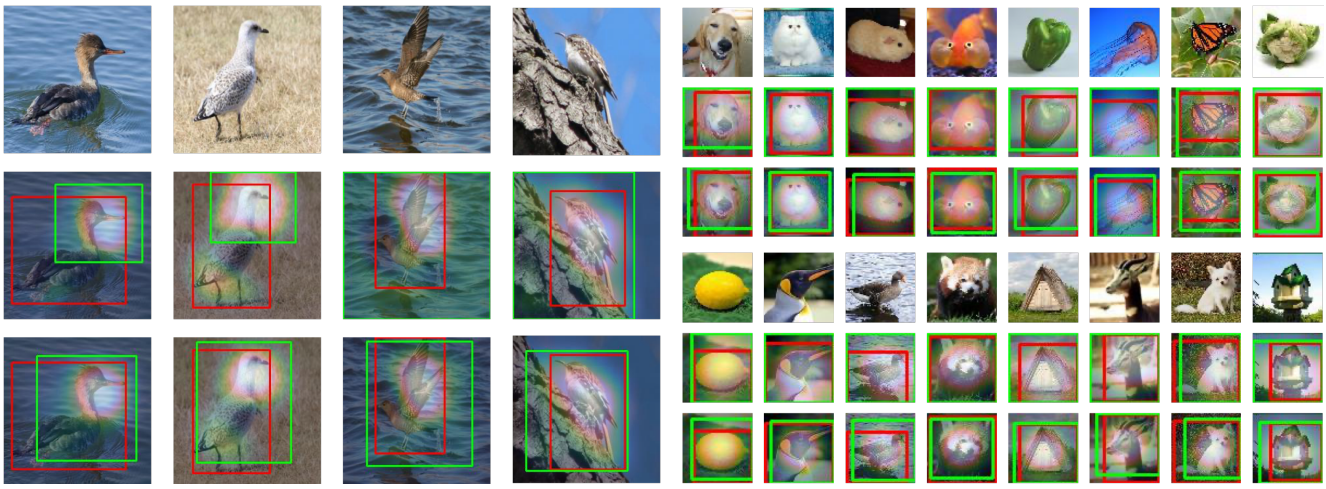

Figure 4: Visualisation of localisation with ResNet50 on CUB-200-2011 and TinyImageNet, without the assistance of ADL. The images in the second row are generated from the original CAM approach and the ones in the third row correspond to infoCAM. The red and green bounding boxes are ground truth and estimations, respectively. Left images are from CUB-200-2011 and right ones come from Tiny-ImageNet. Brighter regions indicate more intense CAM levels.