
RETHINKING SOFTMAX WITH CROSS-ENTROPY: NEURAL NETWORK CLASSIFIER AS MUTUAL INFORMATION ESTIMATOR

A PREPRINT

Zhenyue Qin^{1,*}, Dongwoo Kim^{1,2,*}, and Tom Gedeon¹

¹Australian National University, Australia

²Pohang University of Science and Technology, Republic of Korea

zhenyue.qin@anu.edu.au, dongwookim@postech.ac.kr, tom@cs.anu.edu.au

*Equal contribution and correspondence

ABSTRACT

Cross-entropy loss with softmax output is a standard choice to train neural network classifiers. While it is reasonable to reduce the cross-entropy between outputs of a neural network and labels, the implication of cross-entropy with softmax on the relation between inputs and labels remains to be better explained. We show that training a neural network with cross-entropy maximises the mutual information between inputs and labels through a variational form of mutual information. Our result provides an alternative view: neural network classifiers are mutual information estimators. The new view leads us to develop an informative class activation map (infoCAM). Given a classification task, infoCAM can locate the most informative features of the input toward a label. When applied to an image classification task, infoCAM performs better than the traditional class activation map in the weakly supervised object localisation task.

1 Introduction

Neural network classifiers play an important role in contemporary machine learning and computer vision [16]. Since the emergence of AlexNet [15], much research has been done to improve the performance of neural network classifiers. To overcome the vanishing gradient in deep networks, the residual connection and various activation functions have been proposed [11, 22, 20]. To improve generalisation, better regularisation techniques such as dropout have been developed [26]. To reach better local minima, various optimisation techniques have been suggested [9, 14]. Although many architectural choices and optimisation methods have been explored, relatively fewer considerations have been shown on the final layer of the neural network classifier: the cross-entropy loss with the softmax output.

The combination of softmax with cross-entropy is a standard choice to train neural network classifiers. It measures the cross-entropy between the ground truth label y and the output of the neural network \hat{y} . The network's parameters are then adjusted to reduce the cross-entropy via back-propagation. While it seems sensible to reduce the cross-entropy between the labels and predicted probabilities, it still remains a question as to what relation the network aims to model between input x and label y via this loss function, *i.e.*, softmax with cross-entropy.

In this work, for neural network classifiers, we explore the connection between *cross-entropy with softmax* and *mutual information between inputs and labels*. From a variational form of mutual information, we prove that optimising model parameters using the softmax with cross-entropy is equal to maximising the mutual information between input data and labels when the distribution over labels is uniform. This connection provides an alternative view on neural network classifiers: they are mutual information estimators. We further propose a probability-corrected version of softmax that relaxes the uniform distribution condition.

This new information-theoretic view of neural network classifiers being mutual information estimators allows us to directly access the most informative regions of input with respect to the labels, given classification tasks. The access

to the most informative regions for the labels leads us to develop infoCAM that can locate the most relevant regions for the labels within an image, given an object classification task. Compared to the traditional class activation map, infoCAM exhibits better performance in the weakly supervised object localisation task.

In summary, we outline our contributions as follows:

- The previous view on cross-entropy with softmax only reflects the relationship between the outputs and the labels. We show that with minor modifications to softmax, neural network classifiers then become mutual information estimators. As a result, these mutual information estimators exhibit the information-theoretic relationship between the inputs and the labels.
- We empirically demonstrate that our mutual information estimators can *accurately* evaluate mutual information. We also show mutual information estimators can perform classification more accurately than traditional neural network classifiers. When the dataset is imbalanced, the estimators outperform the state-of-the-art classifier for our example.
- We propose the informative class activation map (infoCAM) which locates the most informative regions for the labels within an image via mutual information. For the weakly supervised object localisation task, we achieve a new state-of-the-art result on Tiny-ImageNet with infoCAM.

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 consist of M classes and N labelled instances as $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $y_i \in \mathcal{Y} = \{1, \dots, M\}$ is a class label of input \mathbf{x}_i . We let $n_\phi(\mathbf{x}) : \mathcal{X} \rightarrow \mathbb{R}^M$ 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 $(\mathbf{X}, Y) \sim P_{XY}$ being 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:

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

Equivalently, following [25], we may express the definition of mutual information in Equation 1 as:

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

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 commonly used lower bound of mutual information via a variational distribution Q [3], derived as:

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

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

The form in Equation 3 is, however, still hard to compute since it is not easy to make a tractable and flexible variational distribution $Q(\mathbf{x}, y)$. Variational distribution $Q(\mathbf{x}, y)$ can be considered as a constrained function which has to satisfy the probability axioms. Especially, the constraint is challenging to model with a function estimator such as a neural network. To relax the function constraint, McAllester *et al.* [21] further apply reparameterisation and define $Q(\mathbf{x}, y)$ in terms of an unconstrained function f_ϕ parameterised by ϕ as:

$$Q(\mathbf{x}, y) = \frac{P(\mathbf{x})P(y)}{\mathbb{E}_{y' \sim P_Y} [\exp(f_\phi(\mathbf{x}, y'))]} \exp(f_\phi(\mathbf{x}, y)). \quad (4)$$

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

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

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

3 NN Classifiers as MI Estimators

In this section, we prove that a neural network classifier with cross entropy loss and softmax output estimates the mutual information between inputs and labels.

To view neural network classifiers as mutual information estimators, we need to discuss two separate cases related to the dataset: whether it is balanced or imbalanced.

3.1 Softmax with Balanced Dataset

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

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

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. In a bid for conciseness, we only provide proof sketches for Theorem 1 and Theorem 2 here. Please refer to the appendix for rigorous proofs.

Theorem 1. *Let $f_\phi(\mathbf{x}, y)$ be $n(\mathbf{x})_y$. Infimum of the expected cross-entropy loss with softmax outputs is equivalent to the mutual information between input and output variables up to constant $\log M$ under uniform label distribution.*

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

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

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

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

which is equivalent to the negative expected cross-entropy loss (7) up to constant $\log M$. Hence, the infimum of the expected cross entropy is equal to the mutual information between input and output variables since the supremum of r.h.s in Equation 5 is the mutual information. \square

y	μ	# samples	$p(y)$
0	0	6,000	0.07
1	+2	12,000	0.13
2	-2	18,000	0.20
3	+4	24,000	0.27
4	-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.

Note that the constant does not change the gradient of the objective. Consequently, the solutions of both the mutual information maximisation and the 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 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' . In experiments, we optimise the revised softmax with empirical distribution on $P(y')$ estimated from the training set. We show the equivalence between optimising the classifier and maximising mutual information with the new softmax below.

Theorem 2. *The mutual information between two random variables X and Y can be obtained via the infimum of cross-entropy with PC-softmax in Equation 10, using a neural network. Such an evaluation is strongly consistent.*

See the proofs in the appendix for the proof of Theorem 2.

Mutual information is often used in generative models to find the maximally informative representation of an observation [12, 31], whereas its implication in classification has been unclear so far. The results of this section imply that the neural network classifier with softmax optimises its weights to maximise the mutual information between inputs and labels under the uniform label assumption.

4 Impact of PC-softmax on Classification

In this section, we measure the empirical performance of PC-softmax as mutual information (MI) and the influence of PC-softmax on the classification task. Since it is impossible to obtain correct MI from real-world datasets, we first construct synthetic data with known properties to measure the MI estimation performance, and then we use two real-world datasets to measure the impact of PC-softmax on classification tasks.

4.1 Mutual information estimation task

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

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

$$P(x|y) = \mathcal{N}(\mathbf{x} | \mu_y, \Sigma_y),$$

where $P(y)$ is a prior distribution over the labels. To form a classification task, we use x as an input variable, and y as a label.

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 in significant proportions of their distributions.

Dimension	Accuracy(%)	Mutual information		
		MC	MINE	softmax
1	74	1.03	1.00	0.99
2	85	1.30	1.22	1.28
5	94	1.54	1.46	1.48
10	98	1.60	1.54	1.54

(a) Results with balanced datasets.

Dimension	Accuracy(%)		Mutual information			
	softmax	PC-softmax	MC	MINE	softmax	PC-softmax
1	79	79	1.02	0.99	1.11	0.96
2	87	88	1.23	1.17	1.31	1.20
5	93	95	1.44	1.27	1.41	1.31
10	95	96	1.48	1.22	1.36	1.34

(b) Results with unbalanced datasets.

Table 2: Mutual information estimation results with softmax-based classification neural networks. MC represents the estimated mutual information via Monte Carlo methods.

We generate two sets of datasets: one with uniform prior and the other with 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 vary the dimension of Gaussian distribution from 1 to 10. The detailed statistics for the Gaussian parameters and the number of samples are available in Table 1. To train classification models, we divide the dataset into training, validation and test sets. 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 mutual information is, however, intractable. We thus approximate it via Monte Carlo (MC) methods using the true probability density function, expressed as:

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

where (\mathbf{x}_i, y_i) forms a paired sample. Equation 11 attains equality as N approaches infinity.

We use four layers of a feed-forward neural network with the ReLU as an activation for internal layers and softmax as an output layer¹. We train the model with softmax on balanced dataset and with PC-softmax on unbalanced dataset. We compare the experimental results against mutual information neural estimator (MINE) proposed in [4]. Note that MINE requires having a pair of input and label variables as an input of an estimator network, the classification-based MI-estimator seems more straightforward for measuring mutual information between inputs and labels of classification 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 access to explicit model parameters for estimating mutual information, whereas the softmax estimator measures mutual information based on the model outputs without accessing the true distribution. We could not find a significant difference between MC and the softmax estimator. Additionally, we report the accuracy of the trained model on the classification task.

Table 2b summarises the experimental results with the unbalanced dataset. The results show that the PC-softmax slightly under-estimates mutual information when compared with the other two approaches. It is worth noting that the classification accuracy of PC-softmax consistently outperforms the original softmax. The results show that the MINE slightly under-estimate the MI as the input dimension increases.

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

Dataset	MNIST		CUB-200-2011	
	Balanced	Unbalanced	Balanced	Unbalanced
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	
	Balanced	Unbalanced	Balanced	Unbalanced
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 for a balanced dataset and are significantly distinct for an unbalanced dataset. Bold values denote p-values less than 0.05 with the Mann-Whitney U test².

4.2 Classification task

We test the classification performance of softmax and PC-softmax with two real-world datasets: MNIST [18] and CUB-200-2011 [29].

We construct balanced and unbalanced versions of the MNIST dataset. For the balanced-MNIST, we use a subset of the original dataset. For the unbalanced-MNIST, we randomly subsample one tenth of instances for digits 0, 2, 4, 6 and 8 from the balanced-MNIST. With CUB-200-2011, we follow the same training and validation splits as in [7]. 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 remaining 6 classes include 29 instances. To construct an unbalanced dataset, similar to MNIST, we randomly drop one half of the instances from one half of the bird classes.

We adopt a simple convolutional neural network as a classifier for MNIST. The model contains two convolutional layers with max pooling layer and the ReLU activation, followed by two fully connected layers with the final softmax. For CUB-200-2011, we apply the same architecture as Inception-V3 [7]. We measure both the micro accuracy and the average per-class accuracy of the two softmax versions on both datasets. The average per-class accuracy 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 unbalanced datasets in terms of the average per-class accuracy.

5 Informative Class Activation Maps: Estimating Mutual Information Between Regions and Labels

In this section, we show that viewing neural network classifiers as mutual information estimators contributes to a more interpretable neural network classifier, via identifying regions of an image that contain high mutual information with a label. There exist previous work exhibiting how to identify regions of an image corresponding to particular labels, known as class activation maps (CAM). We further formalise CAMs to be related to information theory. Furthermore, with the new view of neural network classifiers as mutual information evaluators, we are able to depict the quantitative relationship between the information of the entire image and its local regions about a label. We call our new CAM Informative Class Activation Map (infoCAM), since it is based on information theory. Moreover, infoCAM can also improve the performance of the weakly supervised object localisation (WSOL) task than the original CAM.

To explain infoCAM, we first introduce the concept and definition of the class activation map. We then show how to apply it to the weakly supervised object localisation (WSOL) task.

5.1 CAM: Class Activation Map

Contemporary classification CNNs such as AlexNet [15] and Inception [27] consist of stacks of convolutional layers interleaved with pooling layers for extracting visual features. These convolutional layers result in feature maps. A

²Accuracy with * is higher than the current state-of-the-art [7].

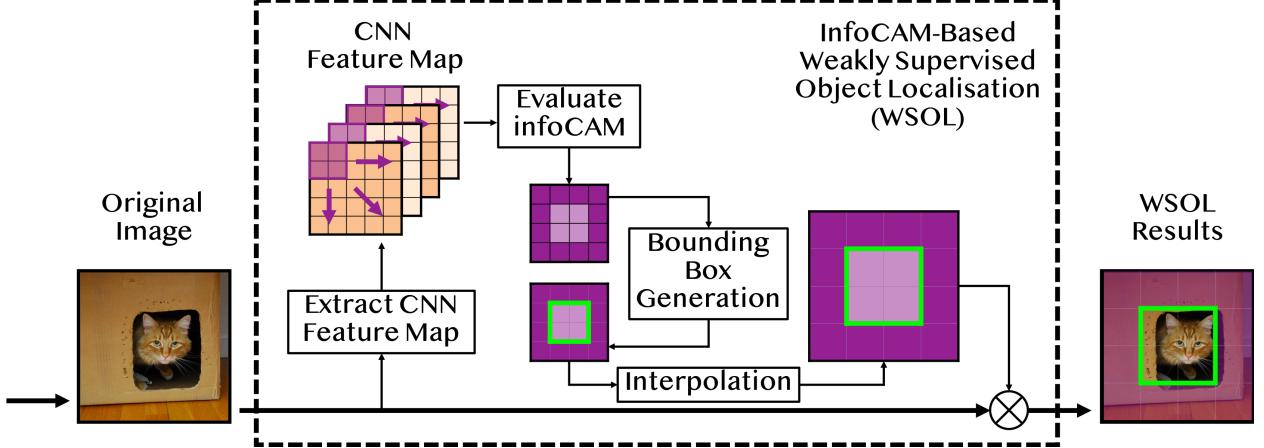


Figure 1: A visualisation of the infoCAM procedure for the WSOL task. The task aims to draw a bounding box for the target object in the original image. The procedure includes: 1) feed input image into a CNN to extract its feature maps, 2) evaluate PMI difference between the true and the other labels of input image for each region within the 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.

feature map is a collection of 2-dimensional grids. The size of the feature map depends on the structure of convolution and pooling layers. Generally the feature map is smaller than the original image. The number of feature maps corresponds to the number of convolutional filters. The feature maps from the final convolutional layer are usually averaged, flattened and fed into the fully-connected layer for classification [19]. Given K feature maps g_1, \dots, 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:

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

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

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

Intuitively, weight w_k^y indicates the overall importance of the k th 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 any pixel-level supervision. Previous approaches tackle the WSOL task by creating a bounding box from the CAM [6]. Such a 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 being a pair of an input image and a label to regions of the input image and labels to capture the regions that have high mutual information with labels.

To simplify the discussion, we assume here 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 introduce a region R containing a subset of grid points in feature map g .

Mutual information is an expectation of the point-wise 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.*,

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

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:

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

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

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

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 an input image into a CNN to extract its feature maps. Then instead of computing the 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 [32]. Finally, we interpolate the generated bounding box to the original image size and merge the two.

6 Object Localisation with InfoCAM

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

6.1 Experimental settings

We evaluate WSOL performance on CUB-200-2011 [29] and Tiny-ImageNet [1]. CUB-200-2011 consists of 200 bird species, including 5,994 training and 5,794 validation images. Each bird class contains roughly the same number of instances, thus the dataset is approximately balanced. Since the dataset only depicts birds, not including other kinds of objects, variations due to class difference are subtle [8]. Therefore, CNN-based classifiers tend to concentrate on the most discriminative areas within an image while disregarding other regions that are similar among all the birds [30]. 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 consists of 500 training and 50 validation images, and 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 Tiny-ImageNet is faster due to image resolution reduction and quantity shrinkage, yet classification becomes more challenging [23].

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 [32]. Specifically, after evaluating infoCAM within each region of an

		CUB-200-2011		Tiny-ImageNet	
		GT Loc. (%)	Top-1 Loc. (%)	GT Loc. (%)	Top-1 Loc. (%)
VGG	CAM	42.49	31.38	53.49	33.48
	CAM (ADL)	71.59	53.01	52.75	32.26
	infoCAM	52.96	39.79	55.50	34.27
	infoCAM (ADL)	73.35	53.80	53.95	33.05
	infoCAM+	59.43	44.40	55.25	34.27
	infoCAM+ (ADL)	75.89	54.35	53.91	32.94
ResNet	CAM	61.66	50.84	54.56	40.55
	CAM (ADL)	57.83	46.56	52.66	36.88
	infoCAM	64.78	53.22	57.79	43.34
	infoCAM (ADL)	67.75	54.71	54.18	37.79
	infoCAM+	68.99	55.83	57.71	43.07
	infoCAM+ (ADL)	69.63	55.20	53.70	37.71

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.

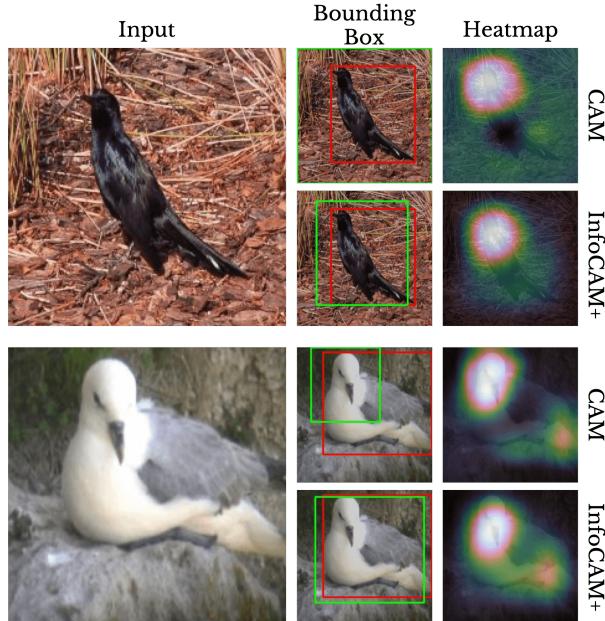


Figure 2: Visualisation of comparison between CAM and infoCAM+. Red and green boxes represent the ground truth and prediction, respectively. Brighter regions represent higher CAM or infoCAM+ values.

image, we only retain the regions whose infoCAM values are more than 20% of the maximum infoCAM and 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.), and 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 predicted most likely image label and also requires correct classification. The localisation of an image is judged to be 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 hyper-parameters as in [6], which shows the current state-of-the-art performance. Specifically, the network backbone is ResNet50 [11] and a variation of VGG16 [27], in which the fully

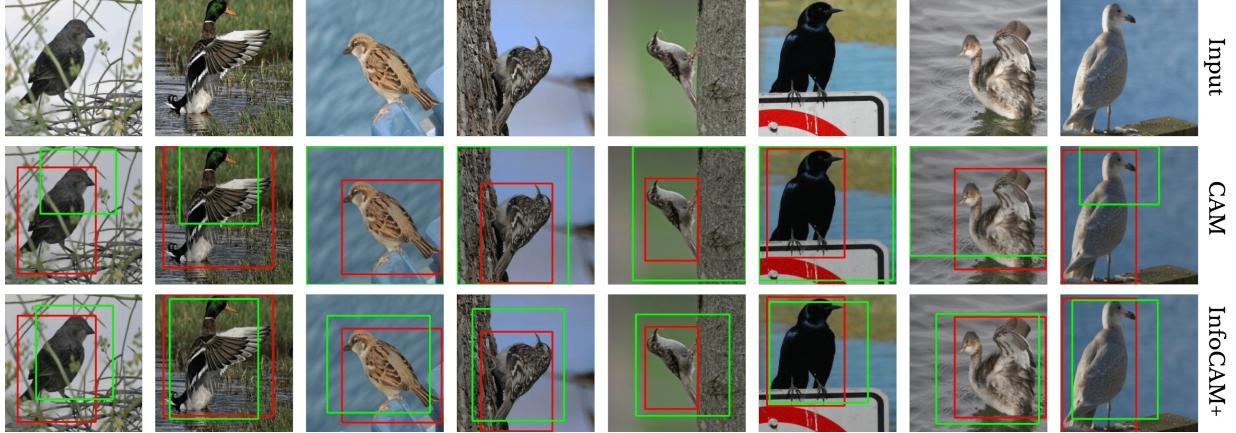


Figure 3: Visualisation of localisation with ResNet50 without using ADL on CUB-200-2011. Images in the second and the third row correspond to CAM and infoCAM+, respectively. Estimated (green) and ground-truth (red) bounding boxes are shown separately.

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

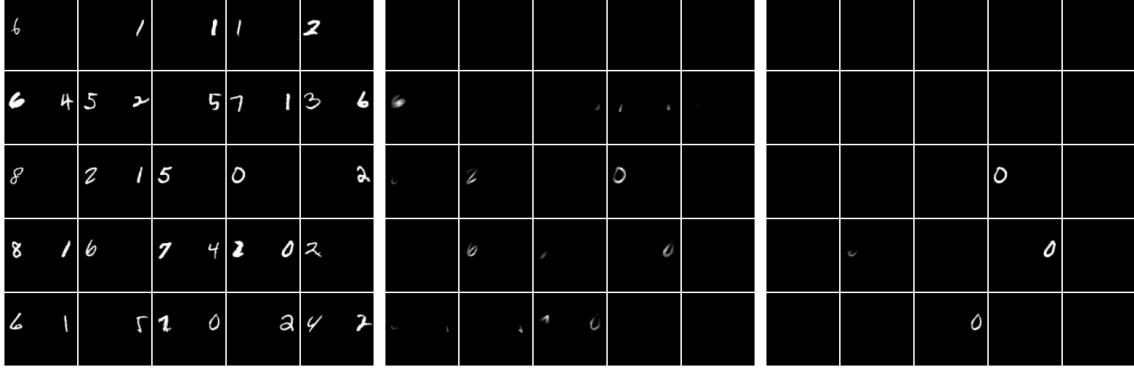
To prevent overfitting in the test dataset, we evenly split the original validation images to two data piles, one still used for validation during training and the other acting as the final test dataset. We pick the trained model from the epoch that demonstrates the highest top-1 classification accuracy in the validation dataset and report the experimental results with the test dataset. All experiments are run on two Nvidia 2080-Ti GPUs, with the PyTorch deep learning framework [24].

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 over the original CAM for WSOL under a wide range of networks and datasets. Both infoCAM and infoCAM+ perform comparably to each other. ADL improves the performance of both models with CUB-200-2011 datasets, but it reduces 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 and 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 9 in the Appendix presents additional examples of visualisation for comparing localisation performance of CAM to infoCAM, both without the assistance of ADL³. From these visualisations, we notice that the bounding boxes generated from infoCAM are formed closer to the objects than the original CAM. That is, infoCAM tends to precisely cover the areas where objects exist, with almost no extraneous or lacking areas. For example, CAM highlights the bird heads in CUB-200-2011, whereas infoCAM also covers the bird bodies.

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 an ablation study to investigate which feature(s) help to localise objects. The results suggest that both components are indispensable to improve the performance of the localisation. For the detailed results, please refer to the ablation study table in the Appendix.



(a) Original input images.

(b) CAM localisation.

(c) InfoCAM localisation.

Figure 4: Visualisation of comparison between CAM and infoCAM for the multi-MNIST dataset. Each image has one or two digits in the left and/or right. We aim to extract digit 0 in each image.

Type	Digit Classification Accuracy (%)									
	0	1	2	3	4	5	6	7	8	9
sigmoid	1.00	0.84	0.86	0.94	0.89	0.87	0.87	0.86	1.00	1.00
PC-sigmoid	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 5: Comparison between the classification accuracy results with sigmoid and PC-sigmoid on the double-digit MNIST dataset.

6.3 Localisation of multiple objects with InfoCAM

So far, we have shown the results of localisation from a multi-class classification problem. We further extend our experiments on localisation to multi-label classification problems.

A softmax function is a generalisation of its binary case, a sigmoid function. Therefore, we can apply infoCAM to each label for a multi-label classification problem, which is a collection of binary classification tasks.

For the experiment, we construct a double-digit MNIST dataset where each image contains up to two digits randomly sampled from the original MNIST dataset [18]. We locate one digit on the left-side, and the other on the right-side. Some of the images only contain a single digit. For each side, we first decide whether to include a digit from a Bernoulli distribution with mean of 0.7. Then each digit is randomly sampled from a uniform distribution. However, we remove the images that contain no digits. Random samples from the double-digit MNIST are shown in Figure 4a.

We first compare the classification accuracy results between using the original sigmoid and PC-sigmoid. As shown in Table 5, PC-sigmoid increases the classification accuracy for each digit type on the test set. InfoCAM improves the localisation accuracy for the WSOL task as well. CAM achieves the localisation accuracy of 91%. InfoCAM enhances the localisation accuracy to 98%. Qualitative visualisations are displayed in Figure 4. We aim to preserve the regions of an image that are most relevant to a digit, and erase all the other regions. From the visualisation, one can see that infoCAM localises digits more accurately than CAM.

7 Conclusion

We have shown the connection between mutual information estimators and neural network classifiers through the variational form of mutual information. The connection explains the rationale behind the use of sigmoid, softmax and cross-entropy from an information-theoretic perspective. The connection also brings a new insight to understand neural network classifiers. There exists previous work that called the negative log-likelihood (NLL) loss as maximum mutual information estimation [2, 17]. Despite this naming similarity, that work does not show the relationship between softmax and mutual information that we have shown here.

The connection between neural network classifiers and mutual information evaluators provides more than an alternative view on neural network classifiers. Via converting neural network classifiers to mutual information estimators, we

³Please refer to the supplementary material for more Tiny-ImageNet visualisation results.

receive two positive consequences for practical applications. First, we improve the classification accuracy, in particular when the datasets are unbalanced. The new mutual information estimators even outperform the prior state-of-the-art neural network classifiers. Second, using the pointwise mutual information between the inputs and labels, we can locate the objects within images more precisely. We also provide a more information-theoretic interpretation of class activation maps. We believe that this opens new ways to understand how neural network classifiers work and improve their performance.

References

- [1] Tiny imagenet visual recognition challenge. <https://tiny-imagenet.herokuapp.com/>. Accessed: 2019-11-03.
- [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.
- [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.
- [4] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540, 2018.
- [5] Junsuk Choe, Joo Hyun Park, and Hyunjung Shim. Improved techniques for weakly-supervised object localization. *arXiv preprint arXiv:1802.07888*, 2018.
- [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.
- [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.
- [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.
- [9] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.
- [10] Sara A Geer and Sara van de Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge University Press, 2000.
- [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.
- [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.
- [13] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [15] 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.
- [16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.
- [17] 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.
- [18] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. 2010.
- [19] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *International Conference on Learning Representation*, 2014.
- [20] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, page 3, 2013.
- [21] David McAllester and Karl Statos. Formal limitations on the measurement of mutual information. *arXiv preprint arXiv:1811.04251*, 2018.
- [22] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. ICML*, 2010.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [27] 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.

- [28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
- [29] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [30] 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.
- [31] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infvae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.
- [32] 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.

Supplementary Materials

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

A Network Architectures

In this section, we illustrate neural network architectures that have been utilised in the previous experiments. Figure 5 demonstrates the architecture of the softmax mutual information neural estimator in section 4. Figure 6 demonstrates the architecture of the network that are utilised to show PC-softmax leads to higher classification accuracy on the unbalanced MNIST dataset as in section 4. We explain in section 6 on how to convert the VGG16 architecture to the VGG16-GAP architecture. Such VGG16-GAP is used in infoCAM. We illustrate in Figure 7 on how to convert the former to the latter architecture. For ResNet50 and Inception-V3, the architectures are identical to [11] and [28].

For more detailed information, please refer to the actual implementation, which we plan to make public.

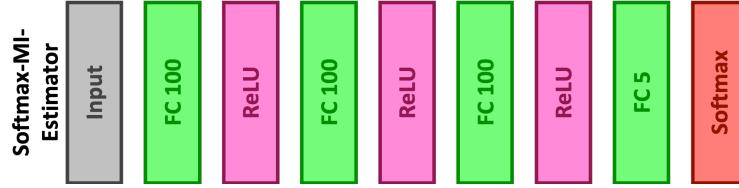


Figure 5: The neural network architecture of the softmax mutual information estimator. The softmax in the last layer can be either the traditional or the PC one.



Figure 6: The neural network architecture that is utilised to show PC-softmax leads to higher classification accuracy on the unbalanced MNIST dataset. The softmax in the last layer can be either the traditional or the PC one.

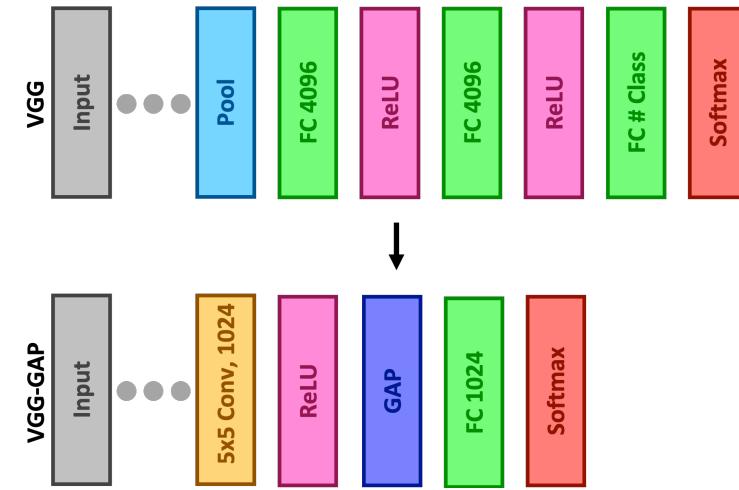


Figure 7: Illustration on the conversion from VGG16 to VGG16-GAP.

B Visualisation of Data Distributions

We show both theoretically and experimentally in section 4 and section 3 that neural network classifiers can be considered as mutual information estimators. In this section, we provide visualisation on the distributions of the data that are used to test the effectiveness of the softmax-based mutual information estimator. In such visualisation as Figure 8, data points are stratified subsets of the test datasets, so that it can reflect the dataset imbalance. Since it is impossible to visualise data whose dimension is greater or equal to three, we apply principle component analysis (PCA) to reduce the dimension to two. Furthermore, data of different class labels become more distinguishable as dimension increases. This can account for the reason why classification accuracy increases as the dimension rises.

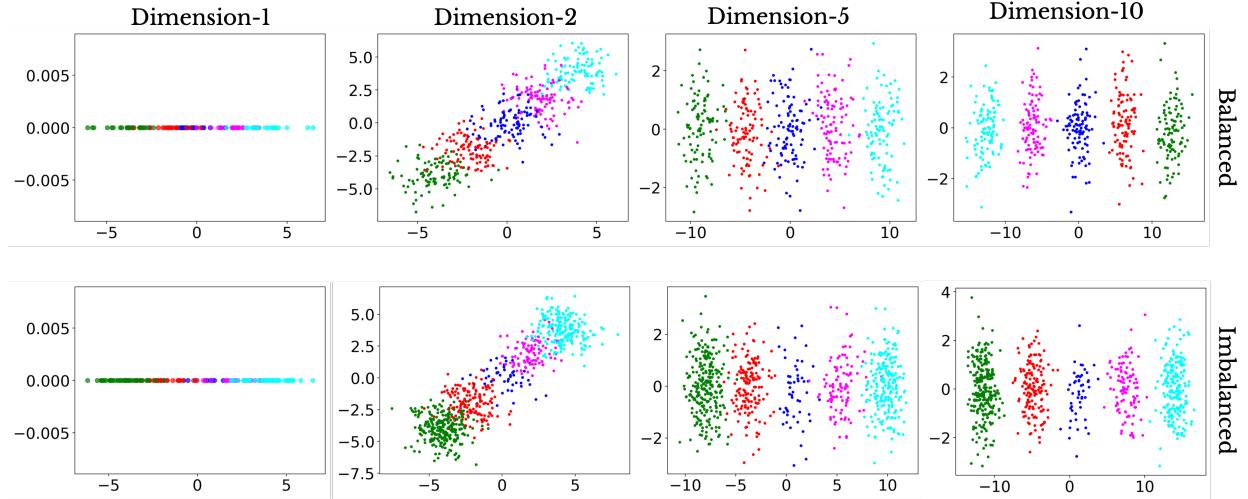


Figure 8: Illustration of the synthetic dataset for evaluating the softmax-based mutual information estimator. For data whose dimension is greater or equal to three, the visualisation is on the results of PCA. The same colour represents the identical class.

C Further Result

In this section, we present some further results on localisation and classification.

C.1 Localisation and Classification Result

Table 7 is a reproduction of main result with the classification results. Note that the classification performances of CAM and infoCAM is the same since we do not modify the training objective of infoCAM. The result can be used to understand the effect of ADL on the classification task.

C.2 Ablation Study

Table 8a shows the result of ablation study. We have tested the importance of three features: 1) ADL, 2) region parameter R and 3) the second subtraction term in Equation 14. To combine the result in the main text, the result suggests that both region parameter and subtraction term are necessary to increase the performance of localisation. The choice of ADL depends on the dataset. We conjecture that ADL is inappropriate to apply Tiny-ImageNet since the removal of any part of tiny image, which is what ADL does during training, affects the performance of the localisation to compare with its application to relatively large images.

C.3 Localisation Examples from Tiny-ImageNet

We present examples from the Tiny-ImageNet dataset in Figure 9. Such examples show the infoCAM draws tighter bound toward target objects.

		GT Loc. (%)	Top-1 Loc. (%)	Top-1 Cls (%)	Top-5 Cls (%)
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	-	-

Table 6: Evaluation results of CAM and infoCAM on CUB-2011-200. 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.

		GT Loc. (%)	Top-1 Loc. (%)	Top-1 Cls (%)	Top-5 Cls (%)
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	-	-

Table 7: Evaluation results of CAM and infoCAM on 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.

D Proofs

In this section, we provide rigorous proofs of Theorem 1 and Theorem 2. The structure of proof is similar to the proof used in [4]. We assume the input space $\Omega = \mathbf{X} \times Y$ being a compact domain of \mathcal{R}^d , where all measures are Lebesgue and are absolutely continuous. We restrict neural networks to produce a single continuous output, denoted as $n(\mathbf{x})_y$. We restate the two theorems for quick reference.

Theorem 1. *Let $f_\phi(\mathbf{x}, y)$ be $n(\mathbf{x})_y$. Minimising the cross-entropy loss of softmax-normalised neural network outputs is equivalent to maximising Equation 5, i.e., the lower bound of mutual information, under the uniform label distribution. That is, if the dataset is balanced, then training a neural network via minimising cross-entropy with softmax equals enhancing a estimator toward more accurately evaluating the mutual information between data and label.*

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. Such an evaluation is strongly consistent.*

The proof technique that we have used to prove Theorem 2 is similar to the one used in [4].

Lemma 1. *Let $\eta > 0$. There exists a family of neural network functions n_ϕ with parameter ϕ in some compact domain such that*

$$|\mathbb{I}(\mathbf{X}; Y) - \mathbb{I}_\phi(\mathbf{X}; Y)| \leq \eta, \quad (16)$$

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 8: 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.

where

$$\mathbb{I}_\phi(\mathbf{X}; Y) = \sup_{\phi} \mathbb{E}_{(\mathbf{X}, Y)} [n_\phi] - \mathbb{E}_{\mathbf{X}} \log \mathbb{E}_Y [\exp(n_\phi)_y]. \quad (17)$$

Proof. Let $n_\phi^*(\mathbf{X}, Y) = \text{PMI}(\mathbf{X}, Y) = \log \frac{P(\mathbf{X}, Y)}{P(\mathbf{X})P(Y)}$. We then have:

$$\mathbb{E}_{(\mathbf{X}, Y)} [n_\phi^*(\mathbf{x})_y] = \mathbb{I}(\mathbf{X}; Y) \quad \text{and} \quad \mathbb{E}_{\mathbf{X}} \mathbb{E}_Y [\exp(n_\phi^*(\mathbf{x})_y)] = 1. \quad (18)$$

Then, for neural network n_ϕ , the gap $\mathbb{I}(\mathbf{X}; Y) - \mathbb{I}_\phi(\mathbf{X}; Y)$:

$$\begin{aligned} \mathbb{I}(\mathbf{X}; Y) - \mathbb{I}_\phi(\mathbf{X}; Y) &= \mathbb{E}_{(\mathbf{X}, Y)} [n_\phi^*(\mathbf{X}, Y) - n_\phi(\mathbf{X}, Y)] + \mathbb{E}_{\mathbf{X}} \log \mathbb{E}_Y [\exp(n_\phi)_y] \\ &\leq \mathbb{E}_{(\mathbf{X}, Y)} [n_\phi^*(\mathbf{X}, Y) - n_\phi(\mathbf{X}, Y)] + \log \mathbb{E}_{\mathbf{X}} \mathbb{E}_Y [\exp(n_\phi(\mathbf{x})_y)] \\ &\leq \mathbb{E}_{(\mathbf{X}, Y)} [n_\phi^*(\mathbf{X}, Y) - n_\phi(\mathbf{X}, Y)] \\ &\quad + \mathbb{E}_{\mathbf{X}} \mathbb{E}_Y [\exp(n_\phi(\mathbf{x})_y) - \exp(n_\phi^*(\mathbf{x})_y)]. \end{aligned} \quad (19)$$

Equation 19 is positive since the neural mutual information estimator evaluates a lower bound. The equation uses Jensen's inequality and the inequality $\log x \leq x - 1$.

We assume $\eta > 0$ and consider $n_\phi^*(\mathbf{x})_y$ is bounded by a positive constant M . Via the universal approximation theorem [13], there exists $n_\phi(\mathbf{x})_y \leq M$ such that

$$\mathbb{E}_{(\mathbf{X}, Y)} |n_\phi^*(\mathbf{X}, Y) - n_\phi(\mathbf{X}, Y)| \leq \frac{\eta}{2} \quad \text{and} \quad \mathbb{E}_{\mathbf{X}} \mathbb{E}_Y |n_\phi(\mathbf{x})_y - n_\phi^*(\mathbf{x})_y| \leq \frac{\eta}{2} \exp(-M). \quad (20)$$

By utilising that \exp is Lipschitz continuous with constant $\exp(M)$ over $(-\infty, M]$, we have

$$\mathbb{E}_{\mathbf{X}} \mathbb{E}_Y |\exp(n_\phi(\mathbf{x})_y) - \exp(n_\phi^*(\mathbf{x})_y)| \leq \exp(M) \cdot \mathbb{E}_{\mathbf{X}} \mathbb{E}_Y |n_\phi(\mathbf{x})_y - n_\phi^*(\mathbf{x})_y| \leq \frac{\eta}{2}. \quad (21)$$

Combining Equation 19, Equation 20 and Equation 21, we then obtain

$$\begin{aligned} |\mathbb{I}(\mathbf{X}; Y) - \mathbb{I}_\phi(\mathbf{X}; Y)| &\leq \mathbb{E}_{(\mathbf{X}, Y)} |n_\phi^*(\mathbf{X}, Y) - n_\phi(\mathbf{X}, Y)| \\ &\quad + \mathbb{E}_{\mathbf{X}} \mathbb{E}_Y |\exp(n_\phi(\mathbf{x})_y) - \exp(n_\phi^*(\mathbf{x})_y)| \\ &= \frac{\eta}{2} + \frac{\eta}{2} = \eta. \end{aligned} \quad (22)$$

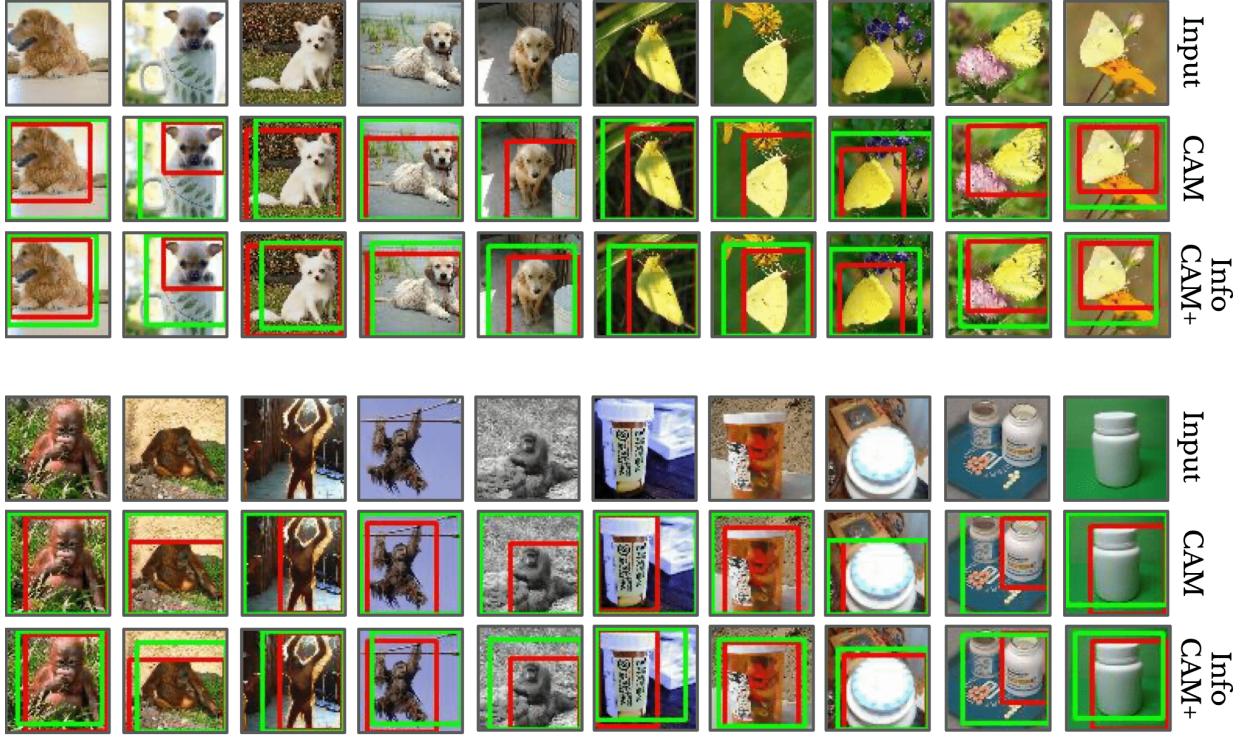


Figure 9: Visualisation of localisation with ResNet50 on CUB-200-2011 and TinyImageNet, without the assistance of ADL. The images in the second row are generated from the original CAM approach and the ones in the third row correspond to infoCAM. The red and green bounding boxes are ground truth and estimations, respectively.

□

Lemma 2. Let $\eta > 0$. Given a family of neural networks n_ϕ with parameter ϕ in some compact domain, there exists $N \in \mathbb{N}$ such that

$$\forall n \geq N, \Pr(|\hat{\mathbb{I}}_n(\mathbf{X}; Y) - n_\phi(\mathbf{X}; Y)| \leq \eta) = 1. \quad (23)$$

Proof. We start by employing the triangular inequality:

$$\begin{aligned} & |\hat{\mathbb{I}}_n(\mathbf{X}; Y) - \mathbb{I}_\phi(\mathbf{X}; Y)| \\ & \leq \sup_{\phi} |\mathbb{E}_{(\mathbf{X}, Y)}[n_\phi^*(\mathbf{X}, Y)] - \mathbb{E}_{(\mathbf{X}, Y)_n}[n_\phi^*(\mathbf{X}, Y)]| \\ & + \sup_{\phi} |\mathbb{E}_{\mathbf{X}} \log \mathbb{E}_Y[\exp(n_\phi)_y] - \mathbb{E}_{\mathbf{X}_n} \log \mathbb{E}_{Y_n}[\exp(n_\phi)_y]| \end{aligned} \quad (24)$$

We have stated previously that neural network n_ϕ is bounded by M , i.e., $n_\phi(\mathbf{x})_y \leq M$. Using the fact that \log is Lipschitz continuous with constant $\exp(M)$ over the interval $[\exp(-M), \exp(M)]$. We have

$$|\log \mathbb{E}_Y[\exp(n_\phi)_y] - \log \mathbb{E}_{Y_n}[\exp(n_\phi)_y]| \leq \exp(M) \cdot |\mathbb{E}_Y[\exp(n_\phi)_y] - \mathbb{E}_{Y_n}[\exp(n_\phi)_y]| \quad (25)$$

Using the uniform law of large numbers [10], we can choose $N \in \mathbb{N}$ such that for $\forall n \geq N$ and with probability one

$$\sup_{\phi} |\mathbb{E}_Y[\exp(n_\phi)_y] - \mathbb{E}_{Y_n}[\exp(n_\phi)_y]| \leq \frac{\eta}{4} \exp(-M). \quad (26)$$

That is,

$$|\log \mathbb{E}_Y[\exp(n_\phi)_y] - \log \mathbb{E}_{Y_n}[\exp(n_\phi)_y]| \leq \frac{\eta}{4} \quad (27)$$

Therefore, using the triangle inequality we can rewrite Equation 24 as:

$$\begin{aligned} |\widehat{\mathbb{I}}_n(\mathbf{X}; Y) - \mathbb{I}_\phi(\mathbf{X}; Y)| &\leq \sup_{\phi} |\mathbb{E}_{(\mathbf{X}, Y)}[n_\phi^*(\mathbf{X}, Y)] - \mathbb{E}_{(\mathbf{X}, Y)_n}[n_\phi^*(\mathbf{X}, Y)]| \\ &+ \sup_{\phi} |\mathbb{E}_{\mathbf{X}} \log \mathbb{E}_Y[\exp(n_\phi)_y] - \mathbb{E}_{\mathbf{X}_n} \log \mathbb{E}_Y[\exp(n_\phi)_y]| + \frac{\eta}{4}. \end{aligned} \quad (28)$$

Using the uniform law of large numbers again, we can choose $N \in \mathbb{N}$ such that for $\forall n \geq N$ and with probability one

$$\sup_{\phi} |\mathbb{E}_{\mathbf{X}} \log \mathbb{E}_Y[\exp(n_\phi)_y] - \mathbb{E}_{\mathbf{X}_n} \log \mathbb{E}_Y[\exp(n_\phi)_y]| \leq \frac{\eta}{4} \quad (29)$$

and:

$$\sup_{\phi} |\mathbb{E}_{(\mathbf{X}, Y)}[n_\phi^*(\mathbf{X}, Y)] - \mathbb{E}_{(\mathbf{X}, Y)_n}[n_\phi^*(\mathbf{X}, Y)]| \leq \frac{\eta}{2}. \quad (30)$$

Combining Equation 28, Equation 29 and Equation 30 leads to

$$|\widehat{\mathbb{I}}_n(\mathbf{X}; Y) - \sup_{\phi} \mathbb{I}_\phi(\mathbf{X}; Y)| \leq \frac{\eta}{2} + \frac{\eta}{4} + \frac{\eta}{4} = \eta. \quad (31)$$

□

Now, combining the above two lemmas, we prove that our mutual information evaluator is strongly consistent.

Proof. Using the triangular inequality, we have

$$|\mathbb{I}(\mathbf{X}; Y) - \widehat{\mathbb{I}}_n(\mathbf{X}; Y)| \leq |\mathbb{I}(\mathbf{X}; Y) - \mathbb{I}_\phi(\mathbf{X}; Y)| + |\widehat{\mathbb{I}}_n(\mathbf{X}; Y) - n_\phi(\mathbf{X}; Y)| \leq \epsilon. \quad (32)$$

□