EXPLORING TARGET DRIVEN IMAGE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

For a given image, traditional supervised image classification using deep neural networks is akin to answering the question 'what object category does this *image* belong to?'. The model takes in an image as input and produces the most likely label for it. However, there is an alternate approach to arrive at the final answer which we investigate in this paper. We argue that, for any arbitrary category $\tilde{y}$, the composed question 'Is this *image* of an object category $\tilde{y}$' serves as a viable approach for image classification via. deep neural networks. The difference lies in the supplied additional information in form of the target along with the image. Motivated by the curiosity to unravel the advantages and limitations of the addressed approach, we propose Indicator Neural Networks(INN). It utilizes a pair of image and label as input and produces a image-label compatibility response. INN consists of 2 encoding components namely: label encoder and image encoder which learns latent representations for labels and images respectively. Predictor, the third component, combines the learnt individual label and image representations to make the final *yes/no* prediction. The network is trained end-to-end. We perform evaluations on image classification and fine-grained image classification datasets against strong baselines. We also investigate various components of INNs to understand their contribution in the final prediction of the model. Our probing of the modules reveals that, as opposed to traditionally trained deep counterpart, INN tends to much larger regions of the input image for generating the image features. The generated image feature is further refined by the generated label encoding prior to the final prediction.

1 INTRODUCTION

Deep neural networks achieve state of the art in supervised classification across different tasks (Rawat & Wang, 2017; Girdhar et al., 2017; Yang et al., 2016). Our work focuses on supervised image classification. Conventionally, while training, the network f_θ is provided as input a set of training images X and corresponding labels Y. It learns by predicting the class labels $\hat{Y} = f_\theta(X)$ and minimising a predefined loss function $\mathcal{L}(\hat{Y}, Y)$. During inference, the network predicts the most likely category for the input image. This approach is analogous to asking a person to name the object present in an image. An alternate approach is to present an image and a class category say cat and ask if the image is of a cat. However, under this scheme one has to exhaustively query every known category to arrive at a final answer. Figure 1 illustrates these scenarios in a natural setting. Prior to the dominance of deep learning based approaches, many methods relied on one-vs-rest SVM(Cortes & Vapnik, 1995) trained on handcrafted image features(Sánchez et al., 2013). The direction saught in this work has a big overlap with the idea of one-vs-rest classification. As we will see in the subsequent sections, we intend to perform a one-vs-rest classification with a single model. To the best of our knowledge, this alternate approach for supervised image classification has not yet been explored in the setting of deep neural networks.

This paper is driven by the curiosity to understand the implications of adopting the plausible alternate strategy of framing the supervised classification task. Our core contributions are as follows:

- We explore an alternate strategy of performing supervised image classification using labels as additional cues for inference. To the best of our knowledge this the first work which provides a unique re-interpretation of the multi-class classification problem.

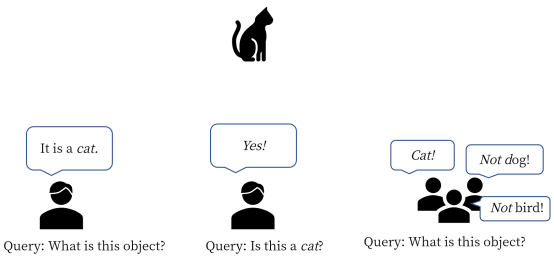


Figure 1: The illustration depicts 3 different approaches of recognising the object of interest. (From left to right) The first person portrays the approach of traditionally trained models. In the middle, is our adopted approach for recognition. The illustration only shows one of the many queries asked. Lastly, individual responses from domain experts represent an ensemble of one-class models.

- To model such a strategy with deep neural networks, we propose a novel architecture termed as Indicator Neural Network(INN). INN produces a binary response conditioned jointly on the input image and query label. It performs multiple one-vs-rest classifications to arrive at the final label assignment.
- Our experiments show that the INNs outperform strong baselines on various image classification datasets. These baselines depict 'traditional' route of training an image classifier.
- We qualitatively and quantitatively investigate the various components of INN and highlight the differences arising due to our pursued structure of the problem.

We have structured the paper as follows: we dive deeper into the motivation behind proposing a new architecture for supervised image classification in section 2. In section 3, we describe the said architecture and it's train and test time methodology. We visit related work w.r.t the proposed architecture in section 4. Section 5 briefly covers the implementation details of the proposed model, selected baselines and chosen datasets. Through sections 6 – 9 we perform various experiments to obtain insights into strengths and weaknesses of the proposed model. We conclude in section 10 by summarising our efforts and discussing the research directions emanating from our work.

2 Motivation For A Novel Architecture

The literature for supervised image classification is vast, as a result, we restrict the discussion to deep learning approaches. The existing solutions for image classification ranging from AlexNet(Krizhevsky et al., 2012) to EfficientNets(Tan & Le, 2019) take the 'traditional' direction for image classification. The traditional direction is depicted in figure 1(left) as a person predicting the category solely based on the input image. These deep learning solutions generate a probability distribution over all known categories as a response and ultimately select the category corresponding to the highest response. The learning of such solutions is backed by categorical cross-entropy loss(Baum & Wilczek, 1988; Solla et al., 1988) which allows a well established framework for training and inference. Other than changing the base architecture, approaches have also been proposed which utilize target transformations(Szegedy et al., 2016; Jarrett & van der Schaar, 2020; Sun et al., 2017), data augmentations(Hongyi Zhang, 2018; Yun et al., 2019) to aid supervised classification. However, these approaches also do not modify the query-response structure of the classifier. Arguably, predictions of a k-way classification model can be interpreted as answering a multi-cue query. This can be achieved by focusing on a single output unit. However, we have to understand that this response is still conditioned only on the input image. Moreover, the learning process ignores the supplied target label.
A recently proposed approach(Khosla et al., 2020) tries to diverge from the norm by utilizing contrastive estimation(Gutmann & Hyvärinen, 2010; Mnih & Kavukcuoglu, 2013) to perform the task of supervised image classification. In a two step process, it first computes an ideal embedding space using positive(images of the same category) and negative(images from other categories) samples. After learning the embedding function, it then trains a traditional classifier(based on cross-entropy loss) on the computed embeddings. The final response however, is yet again an answer to the query 'Which category does this image belong to?' conditioned only on the input image.

As we noted from the above discussion, the existing methods do not provide us with an appropriate way to model supervised predictions conditioned on images and labels. Specifically, allowing us model the query 'Is this image of a *cat*?'. As a result, we propose a novel architecture termed Indicator Neural Networks(INN), which we introduce in the subsequent section.

3 METHOD

We consider a random image-label pair as $(x, \tilde{y})$. We represent a deep neural network, f_θ with learnable parameters θ. Let $\tilde{y}$ represent a one-hot encoded vector of a randomly sampled category. To infer the ground-truth category for an input image, all pairings of image and class categories are required to be queried. The class label corresponding to which a largest response is recorded, it can be assigned as the predicted category for the displayed image. Assuming there are Y' unique labels in the data, this would imply Y' queries for obtaining the predicted category for one image. We model this approach using INNs, $f_\theta(x, \tilde{y})$. The naming is motivated by indicator functions ($\mathbf{1}_{\tilde{\mathrm{y}}=\mathrm{y}}$) as for a single input of image and label, the aim of the model is to predict

$$f_\theta(x,\ \tilde{y}|\ y) = \hat{y} = \begin{cases} 1, & \text{if } \tilde{y} = y \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where, y is the correct label corresponding to x. Realistically an INN will output $\hat{y} \in [0, 1]$.

3.1 INN ARCHITECTURE

We break down f_θ into its components which comprises of an image encoder, label encoder and predictor denoted respectively as:

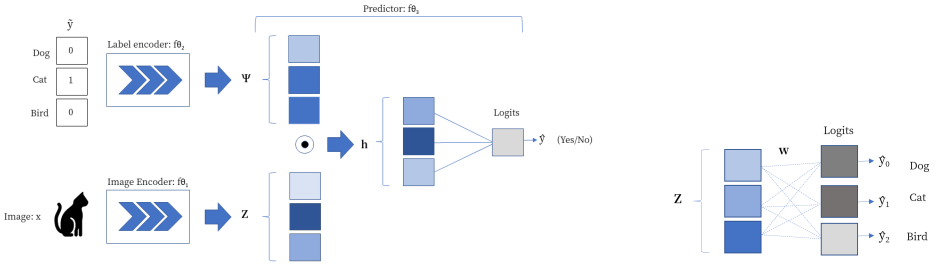
$$\begin{aligned} f_{\theta 1}(x) &= z\ \in \mathbb{R}^d, \\ f_{\theta 2}(\tilde{y}) &= \psi\ \in \mathbb{R}^d, \\ f_{\theta 3}(z,\ \psi) &= \hat{y}\ \in [0, 1]^1. \end{aligned} \tag{2}$$

Here, d represents the dimensions of the embedded features. z and ψ are image and label encodings respectively. Note, that for generating z the input $\tilde{y}$ is irrelevant, and similarly for ψ the input image doesn't matter. The predictor utilises z and ψ to generate the joint image-label representation $h = z \circ \psi\ \in \mathbb{R}^d$. $\circ$ is the element-wise multiplication. It then utilizes h to make the final linear classification decision. Figure 2 shows the pipeline as described above alongside last layers of a traditionally trained model for a visual comparison.

Hypothesis: To have a better understanding of what the model is performing under the hood, we can consider ψ comparable to a $\mathbf{1}^d$ attention map. As a result, ψ will magnify or diminish certain features in z to produce a refined h. We suspect that this reduces the burden of image encoder to produce strong category discriminative features and allows the network to attend to larger regions of the input image. But what stops image encoder from focusing on irrelevant regions in the input image? To answer it, we have to change the perspective with which we observe h. We can also view h as a non-uniformly scaled label embedding(ψ scaled by z). Predictor is necessarily a linear classification head and for it to function appropriately, z extracted from different images of the same category should be similar. As, this will allow the predictor to learn meaningful classification boundaries. As an example, the image encoder will seek common characteristics in all the images of the category *dog*.

3.2 INN TRAINING

To train the INN, we utilise positive and negative pairings of images and labels. The target of the model is to predict *no*(0) for an incorrect pairing whereas, *yes*(1) for a correct one. For a batch of correctly paired input data(sized b), we first extend the batch by concatenating randomly generated incorrect pairings to it. If N is the desired number of incorrect pairings per image per batch, the the resulting size of the input batch after the concatenation operation will be $(N + 1) \times b$. By applying the *i.i.d* assumption for image-label pairs, we can write the empirical log-likelihood which



(a) INN pipeline (b) Single cue recognition model

Figure 2: (a) The proposed 2 stream model for INN. The joint image-label representation(h) is computed from image(z) and label(ψ) representations. This representation is solely responsible for driving the binary prediction of the model. (b) Last layers of a traditionally trained model.

the network aims to maximize as:

$$log(P(\hat{Y}|X,\tilde{Y};\theta)) \implies log(\Pi_{i=0}^{b\times(N+1)} P(\hat{y}_i|x_i,\tilde{y}_i;\theta)) \implies \sum_{i=0}^{b\times(N+1)} log(P(\hat{y}_i|x_i,\tilde{y}_i;\theta)) \quad (3)$$

Alternatively, in terms of loss, for a single image(x), input query label($\tilde{y}$) and ground-truth class label(y) the corresponding loss is denoted as $\mathcal{L}(f_\theta(x,\ \tilde{y}),\ \mathbf{1}_{\tilde{y}=y})$. We employ binary cross-entropy for the implementation of loss. We extend the loss for a single image to the entire dataset as,

$$\mathcal{L}(X,\ Y) = \frac{1}{|X|} \sum_{(x,\ y)\in(X,\ Y)} \left\{ \frac{1}{K_1}\mathcal{L}(f_\theta(x,y),\ 1) + \frac{1}{K_2} \sum_{\tilde{y}_i \in Y' - \{y\}}^{i<N} \mathcal{L}(f_\theta(x,\tilde{y}_i),\ 0) \right\} \quad (4)$$

3.2.1 Comparison to traditional training

It is relevant to point out the differences between an INN and traditional mode of training.

1. Traditionally, the networks designed for supervised classification maximise the likelihood $P(Y|X;\theta)$. In our case, the predictions are conditioned both on the input image and the randomly supplied target.
2. Negative labels are involved indirectly in the loss computation(cross-entropy) due to the softmax operation (Goodfellow et al., 2016, Chapter 6.2.2.3). The supplied target corresponds to the correct label and the resulting contribution to the loss is from the output unit corresponding to this target label. In our framework, the negative classes(stemming from incorrect pairings) are directly involved into the loss computation as we explicitly provide a dedicated target for them.
3. Backpropagating gradient $\frac{\partial \mathcal{L}}{\partial h}\frac{\partial h}{\partial z}$ for the image encoder branch is scaled by ψ due the nature of bi-linear operation. Similarly for label encoder, the gradients are scaled by z. This aspect allows the model to eventually learn compatible representations to make the final prediction.

3.3 INN Inference

For inferring the class label of an input image x, we select the input label which yields the largest response. Formally,

$$\hat{y} = \arg\max_{\tilde{y}} f_\theta(x,\tilde{y}) \ \ \forall \tilde{y} \in Y' \quad (5)$$

4 Related Work

Two-stream models have been deployed successfully for the tasks of action recognition(Simonyan & Zisserman, 2014; Feichtenhofer et al., 2016), video classification(Wang et al., 2018), fine-grained image classification(Lin et al., 2015), multi-label image classification(Yu et al., 2019) and aerial scene classification(Yu & Liu, 2018) to name a few. Apart from the evident difference in the

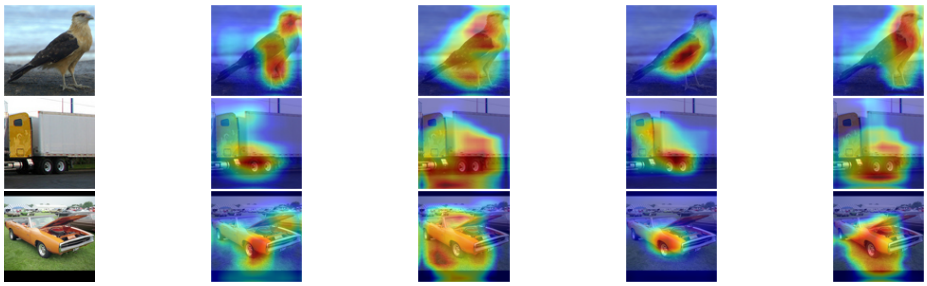
application of these models, the differences lie in the choice of inputs and the function for fusing the 2 stream outputs.
Many approaches have been proposed which utilize labels as auxiliary inputs in image classification (Weston et al., 2010; Frome et al., 2013; Akata et al., 2016; Sun et al., 2017), text classification (Weinberger & Chapelle, 2009; Guoyin Wang, 2018; Dong et al., 2020), and text recognition(Rodriguez-Serrano et al., 2015). In computer vision, these approaches rely on a language model(Mikolov et al., 2013) trained on external data to obtain label embeddings. The main focus of these approaches (Frome et al., 2013; Gang Wang & Forsyth, 2009; Wang & Mori, 2010; Akata et al., 2016) is to use the pre-learnt embeddings to enforce high similarity between image representations of contextually similar categories. These methods are targeted towards zero-shot learning as they rely on enforced similarities to detect novel image categories. As opposed to the existing line of work, we use one-hot encodings as input to our classifier which removes the requirement to utilize any external data. Also, we work without explicitly enforcing similarity constraints on learnt embeddings.
In our training we utilize negative pairing of images and labels. This idea is based on the principle of noise contrastive estimation(Gutmann & Hyvärinen, 2010). SCL(Khosla et al., 2020) also follows this direction to learn meaningful embeddings in their classification approach. Their positive and negative samples consists of images from same and different categories respectively. In contrast, we consider the correctly paired image-label combinations as positives and incorrectly paired image-labels as negatives. Also, ours is a single stage end-to-end differentiable training routine. In INNs, we can assign to label encodings the role of a $\mathbf{1}^d$ attention map(Xu et al., 2015). For image classification, the existing approaches based on attention(Wang et al., 2017; Woo et al., 2018; Hu et al., 2018; Bello et al., 2019; Jetley et al., 2018) introduce spatial or channel-wise attention at different depth of a traditional neural network. In contrast to our proposed model, this modification is made to the image encoder. We can easily replace INN's image encoder with the one equipped with such an attention mechanism. This will incorporate a dual attention mechanism at the level of label fusion and image embedding. However, INN depict one of the simplest ways of modelling the pursued query structure and it is this formulation which gives rise to attention. Attention based approaches as mentioned above focus on answering the query 'What category does the image belong to?'. Moreover, we focus our work to compare different approaches for modelling the classification task rather than different mechanism of performing a traditional classification task.

5 Implementation Details

Datasets: Throughout our paper, we refer to a size of a dataset for the number of unique categories it contains. For small datasets we use CIFAR-10, STL-10, BMW-10(Ultra fine-grain cars dataset), CUB-20(formed using 20 categories of CUB-200-2011), and Oxford-IIIT Pets. Study involving larger dataset utilizes CUB-200. Table provided in appendix A.2 shows the common statistics of the utilized datasets.
Architectures: Here we provide brief details of selected baselines and INN. All the models are trained from scratch to provide an even ground for comparison. Detailed hyper-parameters are provided in appendix D.

- **Baseline-Traditional(B-T)**: We've selected Resnet-18(He et al., 2015) trained with categorical cross-entropy loss as our traditional baseline. It is a widely popular architecture and portrays the standard manner of training an image classifier(Khosla et al., 2020; Tan & Le, 2019). Evaluation with a VGG-11(Simonyan & Zisserman, 2015) model is shared in appendix A.4.
- **Baseline-Multi-Label(B-ML)**: We train the Resnet-18 as a multi-label classifierNam et al. (2014). Each of the Y' output units is treated independently with its own binary cross-entropy computation. This allows us to use $Y' - 1$ output units as negative targets in training.
- **Supervised Contrastive Learning(SCL)**(Khosla et al., 2020): In a much recently proposed approach, the authors make use of contrastive loss based supervised representation learning. As the second step, a linear classifier is trained on top of learnt representations by employing standard cross entropy loss. We train Resnet-18 using the official code[1].

[1]https://github.com/HobbitLong/SupContrast



(a) Original (b) B-T (c) B-ML (d) SCL (e) INN

Figure 3: Grad-CAM visualisations on 'bird', 'car' and 'truck' categories of the STL-10 dataset.

Dataset	CIFAR-10	STL-10	BMW-10	CUB-20	Pets
B-T	**99.82**%	22.67%	11.96%	25.76%	34.83%
B-ML	85.72%	**47.42**%	29.83%	**44.08**%	51.29%
SCL	24.24%	13.25%	4.29%	3.35%	4.19%
INN(N=9)	98.86%	35.06%	**34.03**%	42.21%	**57.37**%

Table 1: Average proportion of pixels considered salient for the training set. The regions considered salient are obtained after applying binarization to Grad-CAM heat maps.

- **INN**: We describe the implementation details of the different components of an INN below.
  - **Image Encoder**: We use a Resnet-18 without the fully connected final layer.
  - **Label Encoder**: We use a 2-layered MLP with no activation(see appendix C.1 for an ablation with activations). The number of units per layer are $d/2$ and d.[2]
  - **Predictor**: z and ψ are combined to form h using element-wise product. h is then connected to the output units which forms the fully-connected final layer for prediction.

6 Experiment: What Does The Network See?

Grad-CAM(Selvaraju et al., 2017) is an approach for interpreting the predictions of a network by qualitatively assessing the identified salient regions in the input image. It utilises the gradient of classification output w.r.t. feature map to generate coarse heatmaps, highlighting important spatial locations in the input image. Recently, Adebayo et al. (2018) assessed different approaches for interpreting a network's prediction. As per their finding, Grad-CAMs generate meaningful heat maps and passed their meticulously constructed sanity tests. Grad-CAM has been utilised by many approaches (Yun et al., 2019; Woo et al., 2018) to emphasize on attended regions by the network. We use Grad-CAM for similar purpose and perform a qualitative and quantitative comparison w.r.t baselines.

Quantitative analysis: Figure 3 shows the heatmaps produced for sample input images for the baseline and INN models. We can notice the significant difference in the spatial spread of salient regions. Comparing the baselines we observe the larger spread on heatmap for B-ML than B-T and SCL. The heatmaps generated for SCL and B-T appear to be localized to highly distinguishable regions. On the other hand, the visuals indicate INN to be looking at a wider region for making a label specific prediction.

Qualitative analysis: To quantify the salient regions we scale the heatmaps between 0 and 1. We consider pixels with values greater than $t = 0.5$ as salient. We use the training set for this comparison. Since we are focused on assessing how the different attended regions vary across methods, the utilization of training data does not restrict us from this goal and moreover, provides us with a larger overlap of accurately predicted samples for computing the salient regions.

Table 1 contain the proportion of an image on an average considered salient as per Grad-CAM. The results are in-line to qualitative assessments we made. For majority of the datasets B-ML and INN produce larger salient regions of the input image. We do not state that focusing on larger regions is

[2]Overall, INN introduces approximately $d \times d/2$ additional parameters. For Resnet-18, $d = 512$

Approach	CIFAR-10	STL-10	BMW-10	CUB-20	Pets
B-T	**95.10**%	86.27%	24.67%	72.23%	70.8%
B-ML	94.66%	88.15%	41.33%	70.29%	66.66%
SCL(Khosla et al., 2020)	93.86%	86.48%	37.40%	68.76%	80.24%
INN($N = 1$)	94.67%	84.83%	36.61%	69.51%	74.24%
INN($N = 3$)	94.42%	85.02%	40.91%	72.03%	78.38%
INN($N = 7$)	94.91%	87.42%	42.12%	73.59%	78.57%
INN($N = 9$)	94.80%	**90.76**%	**44.49**%	**74.36**%	**80.64**%

Table 2: Top-1 accuracy for image classification.

beneficial as compared to more focused distinguishable features. We only aim to support our hypothesis behind the working of an INN. As per our assumption, we hypothesized that the production of disjoint representations z and ψ allows for less discriminative features z. Here we interpreted increase in spatial spread of saliency as producing less discriminative features thereby supporting our hypothesis.

7 Experiment: Image Classification

We evaluate the performance of INNs against small datasets($Y' < 50$). To train INNs, we use $K_1 = K_2 = 1$ as the value of scaling constants in equation 4.

Results: The corresponding results reported in table 2 highlight the effectiveness of INNs. There are four key observations to be made. Firstly, B-T and B-ML show peculiar trend across datasets. In STL-10, B-ML outperforms B-T, we hypothesize that as the predictions are based on a larger input image region which proves beneficial where categories are visually dissimilar. Consequently, for fine-grained visual classification datasets, where the categories are highly similar, B-T performs better.

Secondly, there is a significant difference in performance of the baselines and INN(N=9) for majority of the datasets. For CIFAR-10, the results are comparable. We believe that the small size of the input image does not provide much room for improvement. To verify this, we conduct an experiment in appendix A.5 with images of STL-10 resized to 32×32. We observe a trend of limited improvement for resized STL-10 as we did for CIFAR-10, which supports our theory.

Thirdly, as the value of N increases the performance of INN increases. We believe this is a direct consequence of providing more negative label examples for a given input image during training. By providing many more samples, the network can learn better(more compatible) representations.

Lastly, INN out performs contrastive learning based approach, SCL. For CUB-20 and Pets, we expect further improvement in the performance of INN as the value for N is smaller than the maximum allowed for these datasets.

8 Experiment: Importance of z and ψ

To understand the relevance of z and ψ, we train a linear classifier on top of z in the traditional manner using multi-class cross entropy loss. We compare the accuracy of the model obtained with that of INN. This will help us understand the nature of z as well as improvements made by ψ.

Implementation details: Using the *train* split of the data we gather z^{train} from $f_{\theta 1}$. Note, that the input $\tilde{y}$ chosen is irrelevant for producing z. Next, we train a multi-class logistic regression classifier using stochastic gradient descent on z^{train}, y^{train}. Additional training details are shared in appendix D.8. For inference, we pass the z^{test} to the learnt classifier and record the predicted class. The INN models selected for extracting z corresponds to INN($N = 9$) in table 2.

Results: Table 3 shows the performance of a classifier trained on top of z in comparison to INN($N = 9$). We observe that for image classification datasets of CIFAR-10 and STL-10, the classification performance of the two approaches is highly comparable. However, we observe significant differences for the fine-grained visual classification datasets. We believe that due to high visual dissimilarity between categories in CIFAR-10 and STL-10, obtained z is sufficient to perform the task of classification. However, in fine-grained datasets since the categories are quite visually similar, ψ plays an important role in further refining the representations. These observations are

inline to our hypothesis behind the working of the model. To further highlight the nature of z and ψ we perform additional experiments in appendix C.2.

	CIFAR-10	STL-10	BMW-10	CUB-20	Pets
z	94.42%	90.54%	35.43%	71.48%	78.42%
INN($N = 9$)	94.29%	90.76%	44.49%	74.36%	80.64%

Table 3: Classification using z.

9 Extension to larger datasets

So far, we have observed that the approach of utilizing labels as an additional cue allows to perform the task of multi-class classification. However, the datasets considered only included few unique categories. In this section, we reflect upon the short comings of adopting our pursued approach and subsequently the failures of INNs.

- For smaller datasets, larger the value of N, higher is the classification accuracy. If we extend this logic to larger datasets such as ImageNet(Deng et al., 2009), the best value of N will be close to 1000. Using a traditional batch size(b) of 128 will push the effective batch size to $128,000$, larger than the largest considered for large mini-batch training. methods(Goyal et al., 2017). To counter such large values of N, one can significantly reduce b which in turn will extend the training time from days to months. In order to draw relevant conclusions in a reasonable time frame, we limit the discussions in this section to CUB-200 which contains 200 unique categories.
- **Latent dimension** plays an important in the predictive performance. We conducted experiments on CUB-200 and CUB-20 by varying the latent dimension of the model between $64, 128, 512, 1024$ and observe that impact is more for CUB-200 than CUB-20. The details of the corresponding experiment are described in appendix C.3.
- Large imbalance of positive and negative samples arising as a result of increasing N can destablize an INN training. For similar reasons, we observe B-ML training to collapse as well. We can balance the weights for positive and negative targets by adjusting their contribution to the loss, however, we find that this approach impedes INN performance. As an alternative, instead of training an INN from scratch on larger values of N, one can initialize the weights from an INN trained on a smaller value N', where $N' < N$. By doing this, we find that not only INN(N) surpasses the accuracy of INN(N') but also performs comparable to the baseline. The corresponding experimentation details and results are provided in appendix C.4.

10 Discussion & Conclusion

As opposed to the traditional approach, we explored the applicability of a target driven method. Specifically, we modelled the question 'Does the given image belong to category $\tilde{y}$'. We showed that it is possible to tackle the multi-class classification problem from a non-traditional perspective. Our aim was not to show that the pursued approach is better, rather, we aimed to explore and highlight the pros and cons of this unexplored paradigm. Our approach adapts classical one-vs-rest approach in a modern deep learning setting.

To achieve this goal, we introduced INNs which rely on a pair of input image and target label to produce a response. By inferring exhaustively with all the target categories we arrive at the final decision. Our study involving class activation maps revealed that INNs utilize much larger regions of the input image to generate features. We hypothesize the imposed independence on image embeddings and labels allow the image encoder to tend to larger regions than highly discriminative features from traditional approaches. We also explored the scenarios where learned image features are adequate to learn a traditional classifier on top. This observation was made for cases where the categories are visually dissimilar. Label embeddings refine the coarse image representations immensely for fine-grained tasks. By pitting INNs against strong baselines we were able to highlight the strength of our adopted approach in comparision. The INNs outperformed the baselines on all the datasets($Y' < 50$) considered for image classification and fine-grained image classification.

Additional experiments on Out-of-distribution(OOD, appendix C) and label embedding(appendix B) analysis helps to broaden our understanding following a one-vs-rest setting. OOD analysis shows that INN performs comparable to contrastive learning based SCL. An indicative qualitative result on learnt label embeddings show that similar categories often have nearby label embeddings.
On the down side, we witnessed the difficulties of extending the method to larger datasets. We consider dependency on latent dimension and N the main reasons for this limitation. To make the approach scalable, we believe, constructing a smarter negative sampling approach will be the direction moving forward.
We see numerous avenues for future research. Our proposed direction of training a neural network is comparable to classical one-vs-rest approaches(Sánchez et al., 2013). Due to the sudden outburst and adoption of deep learning approaches, the classical one-vs-rest direction has suddenly phased out. And, to cover and compare all the aspects of a traditionally trained neural network which evolved over the past years in a single work is not feasible. As a result, there are multitude of directions of adopting a one-vs-rest approach as devised in this work. Some directions include but are not limited to object detection(Ren et al., 2015), image segmentation (Chen et al., 2018), anomaly detection(Chandola et al., 2009). Our main focus will be to extend our experimentation theme(and not just the INN) to these problems and analyse its subsequent impact. We will publicly share the source code supplied in supplementary to facilitate brisk research.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 9505–9515. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf.

Zeynep Akata, Florent Perronnin, Zaïd Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1425–1438, 2016.

Eric Baum and Frank Wilczek. Supervised learning of probability distributions by neural networks. In D. Anderson (ed.), *Neural Information Processing Systems*, pp. 52–61. American Institute of Physics, 1988. URL https://proceedings.neurips.cc/paper/1987/file/eccbc87e4b5ce2fe28308fd9f2a7baf3-Paper.pdf.

Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL https://doi.org/10.1145/1541880.1541882.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. JMLR Workshop and Conference Proceedings. URL http://proceedings.mlr.press/v15/coates11a.html.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411. URL https://doi.org/10.1023/A:1022627411411.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Y. Dong, P. Liu, Z. Zhu, Q. Wang, and Q. Zhang. A fusion model-based label embedding and self-interaction attention for text classification. *IEEE Access*, 8:30548–30559, 2020.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 2121–2129. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5204-devise-a-deep-visual-semantic-embedding-model.pdf.

Gang Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *2009 IEEE 12th International Conference on Computer Vision*, pp. 537–544, 2009.

Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. volume 15 of *Proceedings of Machine Learning Research*, pp. 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. JMLR Workshop and Conference Proceedings. URL http://proceedings.mlr.press/v15/glorot11a.html.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2017.

Wenlin Wang Yizhe Zhang Dinghan Shen Xinyuan Zhang Ricardo Henao Lawrence Carin Guoyin Wang, Chunyuan Li. Joint embedding of words and labels for text classification. In *ACL*, 2018.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Workshop and Conference Proceedings. URL http://proceedings.mlr.press/v9/gutmann10a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017.

Yann N. Dauphin David Lopez-Paz Hongyi Zhang, Moustapha Cisse. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Daniel Jarrett and Mihaela van der Schaar. Target-embedding autoencoders for supervised representation learning. 2020.

Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip Torr. Learn to pay attention. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyzbhfWRW.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25, pp. 1097–1105. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 2015.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *International Conference on Computer Vision (ICCV)*, 2015.

Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pp. 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589336. doi: 10.1145/1873951.1874254. URL https://doi.org/10.1145/1873951.1874254.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013.

Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pp. 2265–2273, Red Hook, NY, USA, 2013. Curran Associates Inc.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification — revisiting neural networks. In Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 437–452, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

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. 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. doi: 10.1162/neco_a_00990. URL https://doi.org/10.1162/neco_a_00990. PMID: 28599112.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 91–99. 2015.

Jose A. Rodriguez-Serrano, Albert Gordo, and Florent Perronnin. Label embedding: A frugal baseline for text recognition. *International Journal of Computer Vision*, 113(3), Jul 2015. ISSN 1573-1405. doi: 10.1007/s11263-014-0793-6. URL https://doi.org/10.1007/s11263-014-0793-6.

Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *Int. J. Comput. Vision*, 105(3):222–245, December 2013. ISSN 0920-5691. doi: 10.1007/s11263-013-0636-x. URL https://doi.org/10.1007/s11263-013-0636-x.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pp. 568–576, Cambridge, MA, USA, 2014. MIT Press.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Sara A. Solla, Esther Levin, and Michael Fleisher. Accelerated learning in layered neural networks. *Complex Syst.*, 2(6):625–639, December 1988. ISSN 0891-2513.

Xu Sun, Bingzhen Wei, Xuancheng Ren, and Shuming Ma. Label embedding network: Learning label representation for soft training of deep networks. In *arXiv*. 2017.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, 2019. URL http://proceedings.mlr.press/v97/tan19a.html.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6450–6458, 2017.

Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Yang Wang and Greg Mori. A discriminative latent model of object classes and attributes. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, pp. 155–168, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3642155545.

Kilian Q Weinberger and Olivier Chapelle. Large margin taxonomy embedding for document categorization. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (eds.), *Advances in Neural Information Processing Systems 21*, pp. 1737–1744. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3597-large-margin-taxonomy-embedding-for-document-categorization.pdf.

Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine Learning*, 81(1):21–35, Oct 2010. doi: 10.1007/s10994-010-5198-3. URL https://doi.org/10.1007/s10994-010-5198-3.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057, 2015.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1174. URL https://www.aclweb.org/anthology/N16-1174.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Wan-Jin Yu, Zhen-Duo Chen, Xin Luo, Wu Liu, and Xin-Shun Xu. Delta: A deep dual-stream network for multi-label image classification. *Pattern Recognition*, 91:322 – 331, 2019. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.03.006. URL http://www.sciencedirect.com/science/article/pii/S0031320319301050.

Yunlong Yu and Fuxian Liu. A two-stream deep fusion framework for high-resolution aerial scene classification. *Computational Intelligence and Neuroscience*, 2018:8639367, Jan 2018. ISSN 1687-5265. doi: 10.1155/2018/8639367. URL https://doi.org/10.1155/2018/8639367.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019.

A APPENDIX

A.1 NOTATIONS

Notation	Description
X	Images of a dataset
Y	Ground-truth label corresponding to X
x, y	An image and its corresponding label
Y'	Unique class categories in the dataset
$\tilde{y}$	An input class label.
f_θ	Proposed INN architecture
$f_{\theta 1}$	Image encoder component of an INN
$f_{\theta 2}$	Label encoder component of an INN
$f_{\theta 3}$	Predictor component of an INN
N	Number of incorrect $(x, \tilde{y})$ used per batch per correct pair (x, y) in training.
b	Batch size consisting of correct x, y pairing
z	Output of $f_{\theta 1}$
ψ	Output of $f_{\theta 2}$
h	Joint image-label representation, $\psi \circ z$

Table 4: Notations

A.2 DATASET STATISTICS

Dataset	Y'	#Training	#Test	Image Size
CIFAR-10(Krizhevsky, 2009)	10	50,000	10,000	32×32
STL-10(Coates et al., 2011)	10	5,000	8,000	96×96
BMW-10(Krause et al., 2013)	10	258	254	224×224
CUB-20[3]	20	515	600	224×224
Pets(Parkhi et al., 2012)	37	3,680	3,669	224×224
CUB-200(Wah et al., 2011)	200	5,994	5,794	224×224

Table 5: Datasets

A.3 GRAD-CAM VISUALIZATIONS

We provide more visualisations to compare the recognised salient regions across baselines in figure 4.

A.4 EXPERIMENT: VGG IMAGE ENCODER

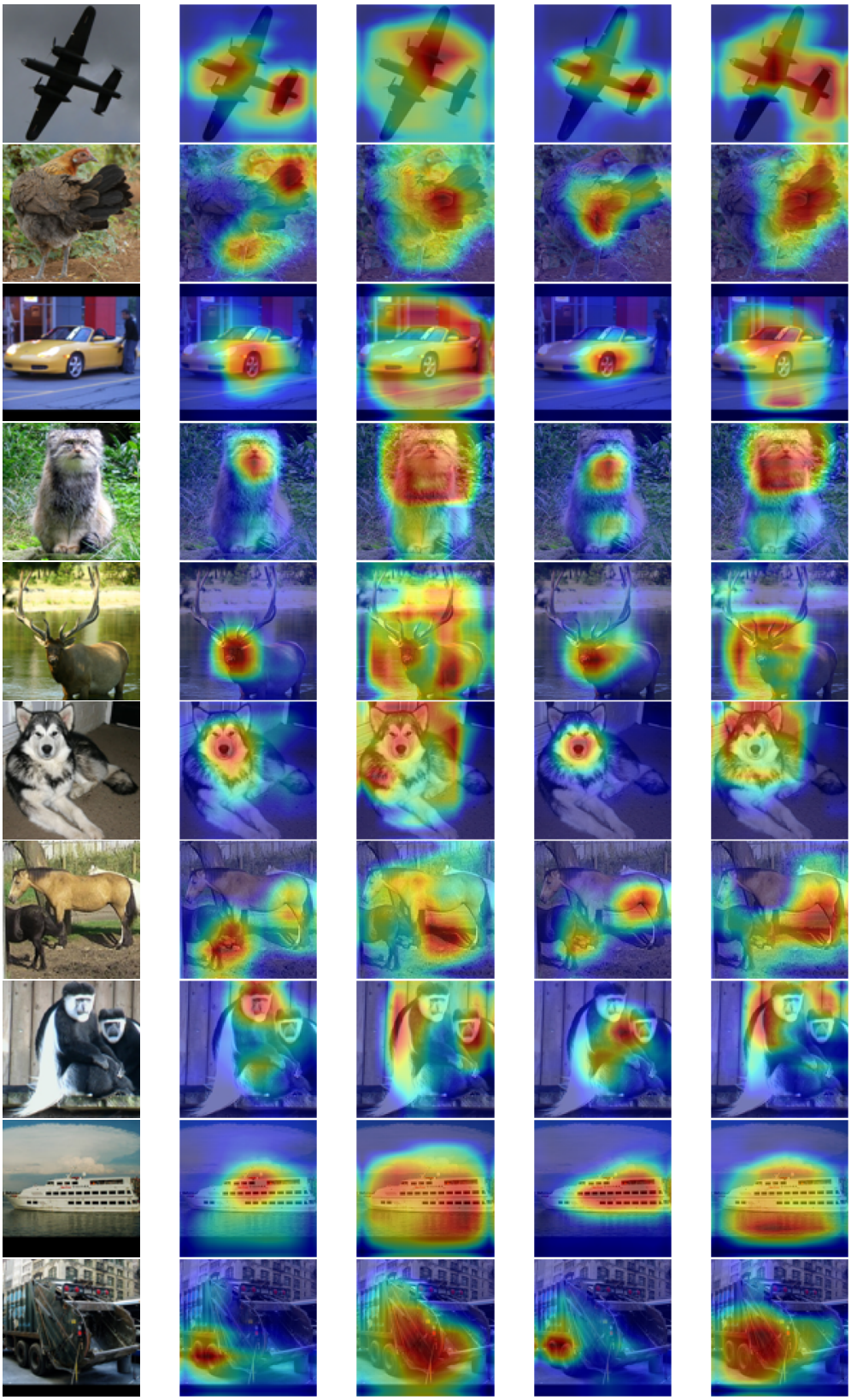
In this section we replace the image encoder of the INN with a VGG-11(with batch normalisation) model. For an INN, we use the features from the last convolutional block after an adaptive average pooling.

Results: Table 6 shows that VGG based INN outperforms the baselines by a large margin. For CIFAR-10, we suspect that similar to the Resnet based INN the small size of the input image restricts the added advantage of using target driven approach.

A.5 EXPERIMENT: RESCALED STL-10

For this experiment, we downscale the STL-10 images to 32×32 to bring it down to the same size as that of CIFAR-10. For training, we use identical hyper-parameters as we did for training the model on unaltered STL-10 dataset.

[3]Created using 20 categories of CUB-200



(a) Original (b) B-T (c) B-ML (d) SCL (e) INN

Figure 4: Grad-CAM visualisations on the training images for the STL-10 dataset. Red region indicates the areas contributing highest towards the prediction of the model.

Approach	CIFAR-10	STL-10	BMW-10	CUB-20	Pets
B-T	91.88%	84.47%	26.3%	68.1%	77.67%
B-ML	92.12%	84.35%	39.7%	68.54%	76.15%
INN($N = 9$)	**92.23**%	**85.57**%	**48.03**%	**74.56**%	**81.62**%

Table 6: VGG-11 Top-1 accuracy

Results: We notice in table 7 that the INN performance is quite similar to that of the baseline when the image size is small. Similar trend was observed in case of CIFAR-10 as well. We believe that INNs and the baseline both utilize equal portion of the input image to generate representations, which leads to similar performance in accuracy.

	32×32	96×96
B-T	78.52%	86.27%
INN($N = 9$)	79.65%	90.76%

Table 7: STL-10 accuracy for different image sizes

B Experiment: Label Embeddings, ψ

We have witnessed that INNs rely on ψ and z to make a correct prediction. Also, depending on the content of the dataset, ψ can play a vital role in further improving the performance. In this experimental set up, we aim to explore more about ψ. Specifically, how different encoded labels relate to each other. We believe that the visual content of images drives the learning of label embeddings, i.e. similar visual categories have nearby label representations. Though the results presented here are qualitative in nature, we believe they provide adequate evidence to back our claim.

Implementation details: We select INN($N = 9$) for CIFAR-10 in this study. We generate $\psi^{Y'} = \{f_{\theta 2}(\tilde{y}) \mid \forall \tilde{y} \in Y'\}$. Next, we compute L2 distance between every pair of entry in $\psi^{Y'}$ as a measure of similarity. In table 8 we have reported the nearest matching labels(smallest distance) for all the categories in the dataset.

Results: Though not perfect, for many source categories, the nearest matching categories tend to be visually similar. For example, the categories truck-car and bird-airplane. However, we also see some non-apparent pairings such as deer-car and frog-car.

Source	Airplane	Bird	Car	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Match	Ship	Airplane	Ship	Dog	Car	Horse	Car	Dog	Car	Car

Table 8: Nearest label match for ψ

C Experiment: Out-Of-Distribution Detection

In this section, we experiment the robustness of the learnt classifiers for detecting out-of-distribution(OOD) images. The standard approach is to utilise the predicted confidence in distinguishing in- and out-of-distribution data(Hendrycks & Gimpel, 2017). Following this framework, we report the AU-ROC for models trained on the chosen datasets while tested on out-of-distribution datasets of LSUN(Yu et al., 2015), Tiny ImageNet(Le & Yang, 2015), Fashion-MNIST(Xiao et al., 2017). The out-distribution datasets are standardised using mean and standard deviation of the in-distribution datasets. The INN models chosen correspond to INN($N = 9$) in table 2.

Results: The results reported in table 9 show that SCL and INN outperform the traditional baselines by a large margin for majority of the datasets. The comparatively lower performance of INN for CUB-20 and Pets can be attributed to its limited training. To recall, the corresponding INNs were trained with $N = 9$, and we expect OOD performance to improve as the values of N used in training is increased.

In	CIFAR-10			STL-10		
Out	LSUN	TIN	F-MNIST	LSUN	TIN	F-MNIST
B-T	**90.3**	86.92	92.2	85.5	79.4	88.75
B-ML	89.14	85.34	92.31	87.5	85.5	89.22
SCL	93.05	**89.9**	**94.2**	75.2	84.72	88.8
INN	87.5	84.6	92.09	**90.7**	**86.6**	**89.79**

In	BMW-10			CUB-20			Pets		
Out	LSUN	TIN	F-MNIST	LSUN	TIN	F-MNIST	LSUN	TIN	F-MNIST
B-T	38.1	35.8	36.9	72.52	70.14	37.7	83.72	74.83	91.04
B-ML	52.5	45.5	35.31	73.94	72.3	59.4	84.0	62.17	90.84
SCL	51.01	52.4	51.82	62.01	**79.34**	**62.01**	**96.9**	**94**	**99.07**
INN	**55.07**	**55.94**	**60.1**	**76.51**	74.64	54.16	92.46	91.76	93.98

Table 9: AUROC(in %) for out-of-distribution evaluation. Higher is better.

RELU	Leaky-RELU	Sigmoid	Tanh	None
90.81%	90.53%	86.5%	90.02 %	90.76%

Table 10: Top-1 accuracy for STL-10 under varying activations

C.1 Experiment: Different activations for label encoder

In the main paper, the label encoder branch consisted of a 2 layered MLP with no activation. In this experiment, we apply the following 4 activations to the label encoder units and train INN($N = 9$, $b = 32$) on the STL-10 dataset.

1. RELU(Glorot et al., 2011)
2. Leaky-RELU(Maas et al., 2013)
3. Sigmoid
4. Tanh

Results: The results indicate maginally better accuracy for RELU and Leaky-RELU. Tanh and *no* activation based models closely follow the accuracy. For sigmoid, the performance is low. Our hypothesis is that, due to the limited scaling nature of the logistic function, the features of z are under refined. However, more extensive research is required to arrive at a stronger conclusion. We hope that our experiment provides an apt working ground for future research in this direction.

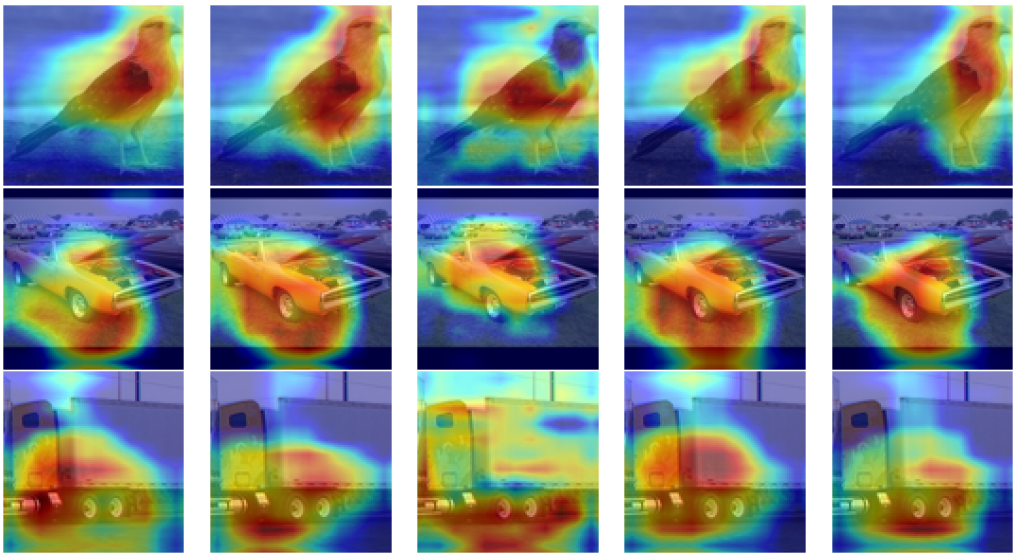
To qualitatively assess the contributing regions of the image across activations, we provide Grad-CAM visualisations in figure 5. RELU, Leaky-RELU, Tanh, and No-activation are able to rely on relevant regions of the input image while making the prediction. In case of Sigmoid, we notice disorganised regions of attention.

C.2 Experiment: Compatibility of ψ & z

To further highlight the fact that INNs do learn compatible representations and rely both on ψ & z to make an accurate prediction, we utilise the following 4 variations of $\tilde{y}$ for evaluating test accuracy on STL-10:

1. $\tilde{y} = y$: We provide the correct class label as input.
2. $\tilde{y} : \tilde{y} \in Y' - \{y\}$: We provide a random incorrect class label as input.
3. $\tilde{y} = \mathbf{1}^{Y'}$: All the values are set to 1 in the input label vector.
4. $\tilde{y} = \mathbf{0}^{Y'}$: All the values are set to 0 in the input label vector.

For evaluation, we record the *argmax* for each individual query between a yes-no response. If the representations are compatible we shall see a higher number of *yes* responses for case 1 than all the other variations.



RELU L-RELU Sigmoid Tanh None

Figure 5: Grad-CAM visualisation for different activations used in the label encoder.

y	$Y' - y$	$\mathbf{1}^d$	$\mathbf{0}^d$
85.2%	0.004%	0.0%	0.0%

Table 11: Test accuracy for different input $\tilde{y}$

Results: Table 11 shows that label encoding ψ play a vital role in classification of the input images. Only when the image is paired with its corresponding ground-truth $\tilde{y}$, INN makes the prediction of *yes* majority of the time. For $\tilde{y}$ corresponding to an incorrect class, the number of samples predicted as *yes* is quite insignificant. For the other two cases, INN never makes a *yes* prediction. This shows that INNs do rely on a compatible z and ψ to generate a correct class prediction.

Visualisation: To further highlight the compatibility of ψ and z we generate a UMAP (McInnes et al., 2018) plot. UMAP is a non-linear dimension reduction technique which has been utilised in visualising high dimensional data. Figure 6 corresponds to the joint representations generated for training images(drawn as blobs) and a single test image of the STL-10 dataset(shown as star). For generating joint representations corresponding to the training set, h^{train}, ground-truth y^{train} are utilised. Whereas, for generating test h^{test}, we provide $\tilde{y} \in Y'$. Consequently, 10 points are generated for a single test image. The ground-truth label of the test image corresponds to *airplane*(integer label of 0).
The figure shows that only when the input label is a one-hot encoded vector corresponding to the ground-truth label *airplane*, h for the test image overlaps with the training cluster(red dashed box). For other input labels, the test sample is further away from its corresponding $\tilde{y}$ cluster.

C.3 Experiment: Varying Hidden Dimension, d

In this experiment we aim to determine the impact of latent dimension on the training of an INN. We conduct this experiment on CUB-200 and CUB-20 datasets with $N = 1$. The latent dimension is selected from the values $\{64, 128, 512, 1024\}$ for a Resnet-18 based INN.
Results: The results in figure 7 indicate the relevance of the dimensions of latent representations. The impact of the latent dimension is more for CUB-200 than CUB-20. For CUB-200 the accuracy increases with increase in dimensionality whereas, for CUB-20, the performance saturates roughly around $d/Y' = 10$ and decreases later on.

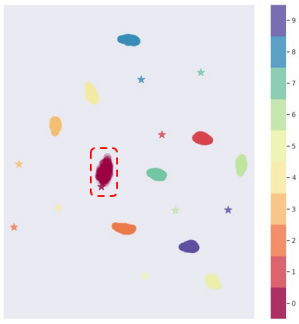


Figure 6: Umap plot

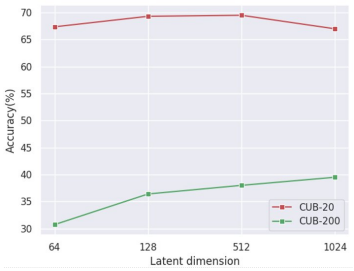


Figure 7: Accuracy vs d

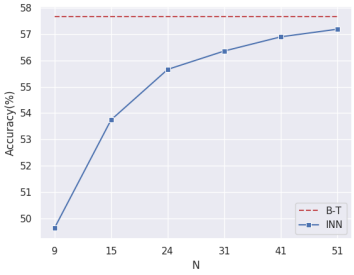


Figure 8: Accuracy on CUB-200

The results indicate that for training larger datasets we are required to employ networks with comparatively larger latent dimensions.

C.4 Experiment: Classification with CUB-200

In order to apply INN to CUB-200 we replace the Resnet-18 image encoder with Resnet-50. The latent dimension is 2048 for Resnet-50. The baseline for this study is B-T. For B-ML, we found that the network doesn't train and obtains an accuracy of 0.5%, which is of a random chance. Even though, INN trains for small values of N, it fails to match its performance on larger values. In order to enable training for an INN when N is large, we initialise the weights from INN(N'), where $N' < N$. For example, we first train the model with $N' = 9$ from scratch and for the subsequent fine-tuning we select the value $N = 15$. If we wish to train on a larger value of N such as 24, we initialize the weights from previously obtained INN($N = 15$). In this study, we select $N \in 15, 24, 31, 41, 51$ and $N' = 9$.
Results: Figure 8 shows the increase in accuracy for an INN with increasing N by applying iterative fine-tuning. The small increment in accuracy at each step is due to proportionally smaller increment of N. $N = 41$ is roughly 20% of the categories of CUB-200. We expect the INN to match and even surpass with higher values of N. However, we did observe the large jump in training time due to lowering of b to accommodate for increasing N. The per epoch time increases from 32 seconds for INN($N = 9$) to 300 seconds for INN($N = 41$).

D Training Details

We firstly cover B-T, B-ML and INN training hyper-parameters. Then we move on to the SCL training hyper-parameters. Baselines(B-T, B-ML) are referred to as N=0 in this section. Deep learning framework used is Pytorch(Paszke et al., 2017) version 1.2.

D.1 CIFAR-10

- Training pre-processing: Random(cropping(32×32, padding=4), rotation(±15), horizontal flipping), normalisation(train mean, std. dev).
- Test pre-processing: Normalisation(train mean and std. dev).
- Epochs: 350
- Start learning rate: 0.1
- Learning rate drop factor: 0.2
- Learning rate drop epochs: 75, 150, 225, 275
- Batch sizes: (N=0, b=256), (N=1, b=128), (N=$\{3, 7, 9\}$, b=64)

D.2 STL-10

- Training pre-processing: Random(cropping(96×96, padding=4), rotation(±15), horizontal flipping), normalisation(train mean, std. dev).
- Test pre-processing: Normalisation(train mean and std. dev).
- Epochs: 350

- Start learning rate: 0.1
- Learning rate drop factor: 0.2
- Learning rate drop epochs: 150, 200, 250, 300
- Batch sizes: (N=0, b=128), (N=1, b=128), (N=3, b=64), (N=$\{7,9\}$, b=32)

D.3 BMW-10

- Training pre-processing: Resized(300×300), Random(cropping(224×224), horizontal flipping), normalisation(train mean and std_dev).
- Test pre-processing: Center Cropping(cropping(224×224), Normalisation(train mean and std_dev).
- Epochs: 350
- Start learning rate: 0.1
- Learning rate drop factor: 0.2
- Learning rate drop epochs: 150, 225, 300
- Batch sizes: (N=$\{0, 1, 3, 7\}$, b=32), (N=9, b=16)

D.4 CUB-20

- Categories: Black_footed_Albatross, Laysan_Albatross, Sooty_Albatross, Groove_billed_Ani, Crested_Auklet, Least_Auklet, Parakeet_Auklet, Rhinoceros_Auklet, Brewer_Blackbird, Red_winged_Blackbird, Rusty_Blackbird, Yellow_headed_Blackbird, Bobolink, Indigo_Bunting, Lazuli_Bunting, Painted_Bunting, Cardinal, Spotted_Catbird, Gray_Catbird, Yellow_breasted_Chat
  These are the first 20 categories as they appeared in torchvision's(Marcel & Rodriguez, 2010) implementation of CUB-200.
- Training pre-processing: Resized(300×300), Random(cropping(224×224), horizontal flipping), normalisation(train mean and std_dev).
- Test pre-processing: Center Cropping(cropping(224×224), Normalisation(train mean and std_dev).
- Epochs: 350
- Start learning rate: 0.1
- Learning rate drop factor: 0.2
- Learning rate drop epochs: 150, 250, 300
- Batch sizes: (N=$\{0, 1, 3, 7, 9\}$, b=32)

D.5 PETS

- Training pre-processing: Resized(300×300), Random(cropping(224×224), horizontal flipping), normalisation(train mean and std_dev).
- Test pre-processing: Center Cropping(cropping(224×224), Normalisation(train mean and std_dev).
- Epochs: 350
- Start learning rate: 0.1
- Learning rate drop factor: 0.2
- Learning rate drop epochs: 150, 225, 300
- Batch sizes: (N=0, b=128), (N=1, b=128), (N=$\{3, 7\}$, b=64), (N=9, b=32)

D.6 CUB-200

- Training pre-processing: Resized(300×300), Random(cropping(224×224), horizontal flipping), normalisation(train mean and std_dev).
- Test pre-processing: Center Cropping(cropping(224×224), Normalisation(train mean and std_dev).
- N=0, Epochs=350, Start learning rate = 0.1, Drop factor = 0.2, Drop epochs=[125, 200, 250, 300], batch size=128
- N=9, Epochs=500, Start learning rate = 0.1, Drop factor = 0.2, Drop epochs=[100, 200, 300, 400, 450], batch size=64
- N=[15, 24, 31], Epochs=300, Start learning rate = 0.005, Drop factor = 0.2, Drop epochs=[100, 200, 250], batch size=[32, 20, 16]

- N=41, Epochs=300, Start learning rate = 0.0025, Drop factor = 0.2, Drop epochs=[100, 200, 250], batch size=12
- N=51, Epochs=300, Start learning rate = 0.001, Drop factor = 0.2, Drop epochs=[100, 200, 250], batch size=10

D.7 SCL TRAINING

Image pre-processing steps are identical to those mentioned in the corresponding previous subsections.
Common parameters: Temperature=0.1, decay(0.0001), cosine(True), and epochs=500.

- CIFAR-10
  - Learning rate: 0.05
  - Batch size: 256
- STL-10
  - Learning rate: 0.5
  - Batch size: 256
- BMW-10
  - Learning rate: 0.1
  - Batch size: 128
- CUB-20
  - Learning rate: 0.5
  - Batch size: 128
- Pets
  - Learning rate: 0.1
  - Batch size: 128

D.8 LINEAR CLASSIFICATION USING z

We have used the SGDClassifier provided by sklearn(Pedregosa et al., 2011) library. Apart from the loss(loss='log') and tol(tol=1e-5) we use the default values to train the model.