# VISUAL EXPLANATION TO UNCERTAINTY VIA GRADIENT BASED LOCALIZATION

*Abhishek Kumar Agarwal*, Arka Mitra*, Krishnam Kapoor*, Prakhar Sharma***

IIT Kharagpur

## ABSTRACT

Deep Neural Networks have achieved an impressive performance over the past decade but they often suffer from poor calibration of their outputs. These networks tend to be over-confident and assign very high probability values to the predicted label. With most networks using softmax at the final layer before calculating prediction probabilities, it leads to a uni-modal distribution of the prediction probabilities and makes it even harder to infer about the confidence of model from the output probabilities. Thus if the model misclassifies an example, it does so with very high confidence. Bayesian Neural Networks can be used to solve this issue of calibration by using the uncertainty of the weights of the model. This helps in inferring situations when the model is not certain about its classification. Hence these models can be deployed safely in critical domains such as healthcare where the misclassifications can be too costly. In this work, we study whether we can successfully use deep Bayesian learning methods to infer uncertainty in a models classification. We especially focus on an evidential learning based deep CNN architecture which uses subjective logic by replacing the output softmax layer with a Dirichlet distribution to model uncertainity. Furthermore, we aim to provide a visual explanation to uncertainity of a model's classification by looking into the region of the image the model looks into and check if they are correlated by using GradCAM. The codes used are publicly available at `https://github.com/thearkamitra/pytorch-classification-uncertainty`.

***Index Terms***— Bayesian Deep Learning, Gradient based localization, Grad-CAM, Explainable AI

## 1. INTRODUCTION

With the rapid growth of the open source Deep Learning frameworks and access to cheap supervised data, the Deep learning methods are starting to become ubiquitous. The performance gains of these methods are huge compared to their traditional rule based counterparts which required domain expertise and were difficult to scale. But these methods have their downside of being inherently black-box in nature, leading to concerns regarding adoption of these methods in critical domains such as healthcare.

Softmax score has been widely used as a confidence measure for DNNs but it tends to give over-confident output. To fix this issue, one line of work considers confidence calibration from a Bayesian point of view [1][2]. In this report we have also investigated this issue. We have tried to use Bayesian methods to infer uncertainty in a deep neural network architecture and how it can be used to re-calibrate model's output probabilities across different datasets. We further use Grad-CAM to generate visual explanations for the outputs of our CNN model. We test our hypothesis that there is a direct correlation between the DNN's uncertainty in classification infered by Bayesian methods and the regions generated by Grad-CAM as the explanations.

The report is structured as follows: Section 2 covers the related work. Section 3, talks about Grad-CAM which is a technique for producing ”visual explanations” for decisions from a large class of CNN-based models, making them more transparent. In section 4, we try explore the correlation between a DNN's uncertainity in classification inferred by Bayesian methods to regions the DNN looks inferred via Grad-CAM.

Our hypothesis is that when Grad-CAM looks at parts of the image which are actually useful for the classification task, the uncertainity of the model should be low whereas when it looks at areas which contain no information relevant to the classification task, the model's uncertainity should be high. We quantify this idea by using two metrics - dice index and IoU (Intersection over union). Finally in section 5 we state the conclusion and scope of improvement in this project.

## 2. BAYESIAN METHODS TO INFER UNCERTAINITY

### 2.1. Varitional Inferencing of Posterior

When the posterior of a deep neural network is inferenced instead of using optimisation methods we term it as Bayesian neural networks (BNN). Deep neural networks tend to overfit in smaller datasets and here comes the utility of BNN's. To use CNN in small size data by applying bayesian approximate inference is strenous as posterior integral is difficult to calculate. Hence, the posterior is modeled using bernoulli variational distribution, and fit the distribution's parameters by minimizing the Kullback-Leibler (KL) divergence from the

*All the authors contributed equally.

true posterior [3]. Dropouts are approximated to the use of bayesian posterior inference of BNN's. Bayesian neural networks can be modelled by just applying dropout after convolution in conventional way. In this way the convolution and pooling layers are computed in a probabilistic way after which dropouts are put into effect. Predictive posterior distribution is helpful in computing average stochastic forward passes during testing. The following equation is the predictive distribution for a given new input say x*.

$$p(y^*|x^*, X, Y) = \int p(y^*|f^*)p(f^*|x^*, \omega)p(\omega|X, Y)df^*d\omega \tag{1}$$

The $\omega$ in the given equation is our parameter variable.The integral is not analytically solvable. Hence the technique of variational inferencing is helpful which uses simple distribution such as bernoulli or gaussian to approximate the posterior.

$$q(y^*|x^*) = \int p(y^*|f*)p(f*|x*, \omega)q(\omega)df*d\omega, \tag{2}$$

q($\omega$) in the given equation is variational distribution. Now KL divergence can be maximised for the integral. The below equation shows lower evidence bound which needs to be minimised for approximation.

$$L_{VI} = \int q(\omega)p(F|X, \omega)\log p(Y|F)dFd\omega - KL(q(\omega)||p(\omega)) \tag{3}$$

Variational dropout is being used in [4] where Gaussian is used instead of Bernoulli as variational distribution. Gaussian random noise is being added in weights. In variational dropout method, each weight is given its own dropout rate. Sparse variational dropout is created by a new approximation of the KL-divergence term in the Variational Dropout objective that is tight on the full domain. The Sparse models thus created are very efficient and fast with reasonable performance. The efficiency of stochastic gradient-based variational inference with mini-batches of data can still be improved if uncertainty in global parameters can be transferred into local noise that is independent across data points[5].

### 2.2. Evidential Learning

The Dempster–Shafer Theory of Evidence (DST) assigns belief masses to each possible class labels for a sample[6]. Subjective Logic (SL) formalizes DST's notion of belief assignments as a Dirichlet Distribution[7]. In SL we consider K mutually exclusive classes and assign a belief mass $b_k$ for each class k = 1, ..., K and an overall uncertainty mass of u. These mass values sum up to one and are all non-negative

$$u + \sum_{k=1}^{K} b_k = 1 \tag{4}$$

These belief masses $b_k$ are calculated using evidence for the class. Let $e_k \geq 0$ be the evidence then the belief $b_k$ and uncertainty $u$ are calculated as

$$b_k = \frac{e_k}{S} \quad and \quad u = \frac{K}{S} \tag{5}$$

where $S = \sum_{i=1}^{K}(e_i + 1)$. Evidence here is the measure of amount of support collected from data in favor of a sample to be classified into a certain class. When there no evidence for any label, the belief for each label becomes zero and uncertainty becomes 1. Now a Dirichlet distribution with parameters $\alpha_k = e_k + 1$ can be used to derive this belief mass assignment with $b_k = \frac{\alpha_k - 1}{S}$ where $S = \sum_{i=1}^{K} \alpha_i$

Normally, a standard neural network's output is used to calculate the probability over the classes but when we use it as evidence for parameterising Dirichlet distribution, it allows to capture the density of probability assignment of classes. [7]

## 3. VISUAL EXPLANATION OF IMAGES

Convolutional Neural Networks result in state of the art results in many tasks like classification [8], object detection [9] and image segmentation [10]. However, we do not get an exact interpretation of why the model is giving the specified output. Thereby, these neural network models are usually treated as black-boxes.

Our method gives an estimation of the uncertainty on the predictions of the model. However, we still are not able to understand why the model is specifying the same. There are method that determines the region of the image the model is at. We tried with two methods, Gradient-weighted Class Activation Mapping (GradCAM) and RISE. However, on the MNIST dataset, they give similar heatmaps for many images so GradCAM is selected as it is faster.

In our case, we had an image classification task. The images are passed through several layers of the model and this activates the features map of that layer. These can give us a sense of which regions of the image the model looks at to give the output. Gradients of the output with respect to the layers in the network tells us how the changes in the layers with respect to the output result in a change in the prediction.

Since we are interested in knowing the regions of the image that are getting activated, the activations in the feature maps are quite important. In a particular layer, there are many feature maps that are present. However, all of them do not contribute equally to the prediction. The feature maps that are more important for the predictions thus need to be given more importance. The weight that is given to the $k_{th}$ layer of the feature maps is given by:

$$\alpha_k = \sum_i \sum_j \frac{\partial y}{\partial A_{i,j}^k}. \tag{6}$$

The weighted average of the individual layer thus leads to a new feature image. We wish to see the regions that have a positive impact of the image, and so we only consider the positive part of the feature image. This has an size which is equal to the feature map size which might not be equal to the original input size image. A resizing is done so that the images are of the same shape. The image before the reshaping is given by:

$$ReLU(\sum_k \alpha_k A^k)$$

The highlighted regions show the regions of the image the model is focussing on as can be seen in the Fig. 1

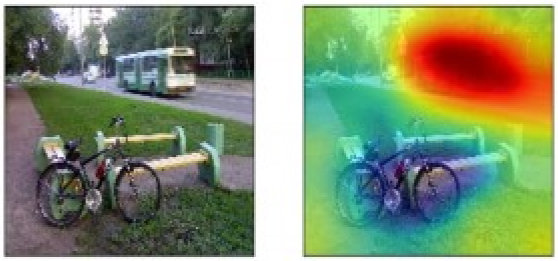


**Fig. 1**: GradCAM for interpreting label bus

## 4. METRICS

We wanted a quantitative description of the regions of the image focused by the model vs the regions where information about the object to be classified lies. This can be achieved by looking at the intersection of the regions highlighted by Grad-CAM and the actual object in the image. For instance, when the image of handwritten digit 8 from MNIST dataset is fed to our LeNet network, we see that the parts highlighted by Grad-CAM exactly corresponds to the region where 8 is written. This is shown in Figure 6.

On the other hand if we rotate this image by 90 degrees and then feed it to our LeNet network, we see that the region highlighted by Gradcam does not correspond to the region where 8 lies. This is shown in Figure 7.

In order to quantify this idea, we use two metrics in this project.

1. **Dice Index:** Dice Index or Dice Similarity Coefficient (DSC) is defined as twice the shared information in two sets upon the sum of cardinalities of the two sets. Its is given as:

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{7}$$

where $|X|$ and $|Y|$ are the cardinalities of X, Y.

2. **Intersection Over Union (IoU):** Intersection over union is popularly used to measure the accuracy of object detectors in machine learning literature. It measures the overlap between the region of ground truth and the region predicted by a classifier. In our case, we take the intersection of the region containing the object with the region highlighted by Grad-CAM. IoU can be given as:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{8}$$

where A and B are the two sets whose intersections we are interested in.

## 5. DATASETS

There are a big range of image datasets that are available today, differing in the number of the images and the size of each image. Due to the small duration of the project and lack of sufficient compute power, we have used the following datasets in our experiments:

- **MNIST:** The MNIST database [11] was constructed from NIST's Special Database 3 and Special Database 1 which contain binary images of handwritten digits. The MNIST training set is composed of 30,000 patterns from SD-3 and 30,0The MNIST dataset consists of binary images of handwritten digits. NIST's Special Database 3 and Special Database 1 was used in the construction of this dataset. It consists of a total of 60,000 training images (30,000 from SD-3 and 30,000 from SD-1) and 10,000 test images (5,000 from SD-3 and 5,000 from SD-1). The samples were obtained from a pool of around 250 writers. Each image is of size 28x28. The dataset has been preprocessed to position the center of mass of the pixels of each image in the center of the 28x28 field. This dataset is widely used as a benchmark for classification problems.

- **Fashion MNIST:** This dataset is similar to MNIST dataset. Instead of handwritten digits, this dataset consists of 28x28 images of apparels from Za-lando's articles[12]. It consists of 10 classes with each example belonging to exactly one class. The size of the training set is 60,000 and the size of the test set is 10,000. The images are binarized and preprocessed to center align in the 28x28 view

- **CIFAR10:** CIFAR10 dataset[13] consists of 10 classes representing cars, cats, birds, deer, horses, ships, trucks, airplanes, dogs and frogs. It is one of most widely datasets for computer vision applications. It was created by Canadian Institute for Advanced Research and its consists of 6,000 images per class. Each image is resized to 32x32 and students were paid to label these images. It is a subset of a larger dataset consisting of 80 million images.

| Dataset | DSC Normal | DSC Rotated | P value |
|---|---|---|---|
| MNIST | 0.21 | 0.35 | 3.53e-20 |
| Fashion MNIST | 0.24 | 0.31 | 7.56e-20 |

**Table 1**: Results on Dice Index

| Dataset | IoU Normal | IoU Rotated | p value |
|---|---|---|---|
| MNIST | 0.12 | 0.24 | 6.29e-13 |
| Fashion MNIST | 0.15 | 0.21 | 1.01e-15 |

**Table 2**: Results on IoU

## 6. MODELS USED

The datasets in the models were trained using deep learning networks. The details of the model are mentioned in details below.

1. **LeNet**: Lecun et al. [14] uses LeNet to classify the MNIST dataset. We use the same architecture, only the activation function at the end has been modified. There are 2 convolutional layers followed by max-pooling. The feature maps are flattened and they pass through a linear layer with ReLU activations. Dropout is known to increase accuracy and thus it is applied after this layer, in contrast to the original model. The layers are passed through a final linear layer. No activations were applied before the softmax function which is also in contrast to the original paper.

2. **ResNet**: We used CIFAR10 dataset for comparision as well. However, the accuracy on that dataset using LeNet was quite low. Deeper networks tend to rise the accuracy on a more difficult dataset. It gets harder to train the network as the network gets deeper. He et al. [15] has shown that adding skip connections in the architecture significantly improves the performance of the model. Thus we used ResNet18 with pretrained weights to get a better metric. ResNet18 has 18 layers 1024 features at the end of the convolutional layers. That is directly connected to a feed forward network.

## 7. RESULTS AND DISCUSSION

### 7.1. MNIST

We train our LeNet5 network using evidential learning as explained in section 2.2. As shown in figure our network classifies an input image of handwritten digit 6 correctly when the rotation is 0 degree. As we rotate the image, the uncertainty of classification increases and hence the network refuses to classify instead of confidently predicting wrong output as observed in case of softmax classifier. After rotating for around 150 degrees, the network classifies the digit as 9. This is in coherence with our expectations since flipping 6 transforms it into digit 9. We also tested our LeNet network on out of sample data (image of one of the authors). As expected, a soft max variant gives unrealistically high classification probabilities and classifies the test image as digit 4. This is shown in 3. On the contrary, replacing the soft max layer with a Dirichlet distribution reduces this overconfidence and the network gives a very high uncertainty value for this out of sample image. This is shown in Figure 4.

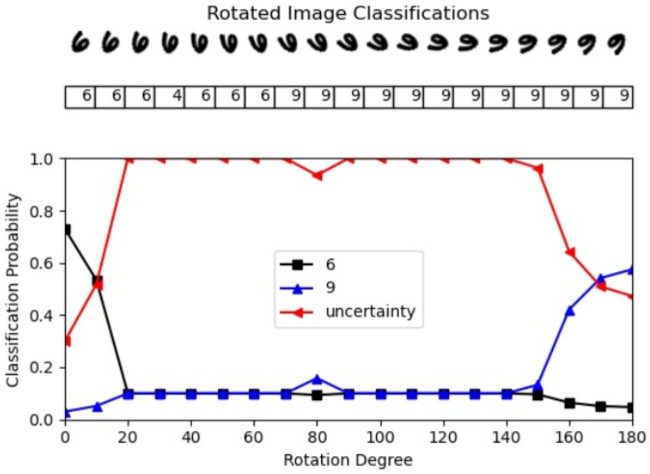


**Fig. 2**: Classification of Hand Written Digit 6

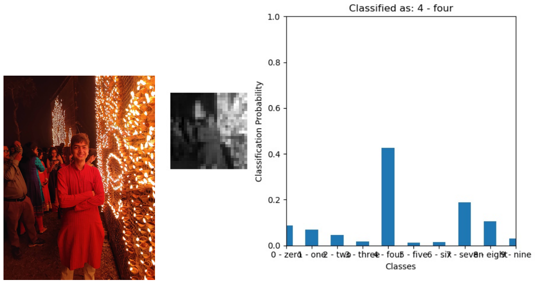


**Fig. 3**: Out of sample image classified as 4 with high probability

### 7.2. Results on Other Datasets

We obtained the best results on MNIST dataset (98.3%). This was followed by Fashion MNIST(95.7%). In case of CIFAR10(71.8%), LeNet did not perform well and hence we trained a ResNet (93.3%) network for this dataset.

### 7.3. Correlating Grad-CAM output with Uncertainty

The out of sample images generated very low probabilities assigned to each class. Thus opted us to see if the regions that

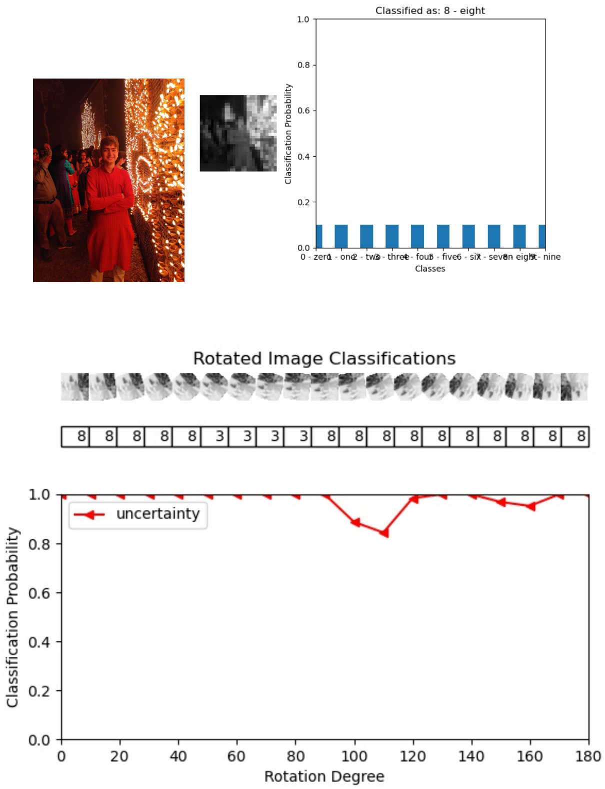


**Fig. 4**: The classifier gives very high uncertainty on out of sample image

the model looks into for this images differ or not. The GradCAM explaination from the model which did not consider uncertainties was shown in Figure 6 while that for model which considered the uncertainties is showed in Figure 7. This opted us to look into the metric stated in Section 4. An image with high certainty has very high value with the dice index and a one which high uncertainty has very low dice index.

### 7.4. Results on the Correlations

The correlation with one image opted us to try it for a batch of images. We ran the experiments on MNIST and Fashion-MNIST and the values that were recorded for both the models containing certainty and not containing certainty. The p-value of the values were taken and it came out to be very low (in the order of $(10^{-21})$). Consequently the hypothesis was rejected. This is shown in Tables 1 and 2.

### 7.5. Reason for Failure

The failure of the hypothesis prompted us to look into more images and understand why the metrics did not work. Subjective viewing of the images showed that the model with high uncertainty looked into different regions that those with low uncertainty. But as shown in Figure 5, the result in the Grad-CAM are matching for with what we would expect for a sample from distribution. The red region corresponds to heatmap for the image. However, due to the shift of the heatmap towards the left, the dice index becomes zero. This might have happened for many images that we had tested and led to a very low p value. We need to find a better metric to match the image and the GradCAM output for that image.

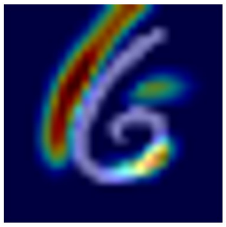
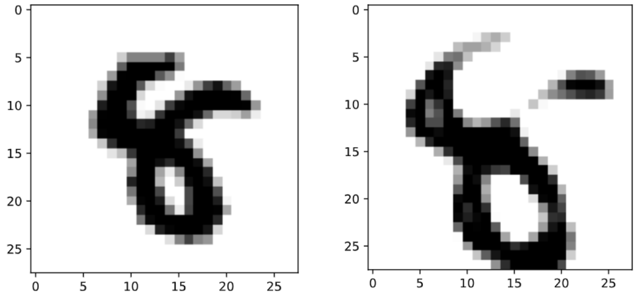
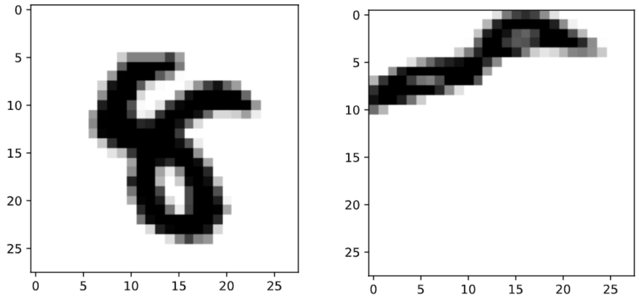


**Fig. 5**: GradCAM results shifted towards right



(a) Input (b) GradCAM Output

**Fig. 6**: The GradCAM on a normal 8.



(a) Input (b) GradCAM Output

**Fig. 7**: The GradCAM on a rotated 8.

## 8. FUTURE WORK

As discussed in the section above, our hypothesis proved to be incorrect for the used data sets. As discussed in section 7.5, this can be a result of choosing the evaluation metric poorly. Hence, we leave it to future studies to try out the described approach on larger data sets and use better metrics to prove or disprove our stated hypothesis.

## 9. REFERENCES

[1] Jost Tobias Springenberg, A. Klein, Stefan Falkner, and F. Hutter, "Bayesian optimization with robust bayesian neural networks," in *NIPS*, 2016.

[2] Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan, "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness," 2020.

[3] Yarin Gal and Zoubin Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," 2016.

[4] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov, "Variational dropout sparsifies deep neural networks," 2017.

[5] Diederik P. Kingma, Tim Salimans, and Max Welling, "Variational dropout and the local reparameterization trick," 2015.

[6] Arthur P. Dempster, *A Generalization of Bayesian Inference*, pp. 73–104, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[7] Audun Jøsang, *Principles of Subjective Logic*, pp. 83–94, Springer International Publishing, Cham, 2016.

[8] Mingxing Tan and Quoc V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020.

[9] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[10] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos, "Image segmentation using deep learning: A survey," 2020.

[11] Yann LeCun and Corinna Cortes, "MNIST handwritten digit database," http://yann.lecun.com/exdb/mnist/, 2010.

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

[13] Alex Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.