Expert Evaluation and Inherited Uncertainty in the Development of CNNs for Damage Recognition

Maria Pantoja, Robert Kleinhenz

May 2018

1 Abstract

Neural Networks (NN) are currently being used for different computer vision tasks; from labeling cancerous cells in medical images to identify traffic signals in self-driving cars. The goal of supervised NN is to classify raw input data according to the patterns learned from an input training set, this input training data is mostly manually labeled. The level of expertise of the professionals labeling the training set sometimes varies widely, also some of the images used are not that clear and are difficult to label, which leads to data sets containing pictures labeled differently by different experts or that contain uncertain labels. These kind of errors on the training set do happen more frequently when the NN task needs to classify numerous labels with similar characteristics, for example when labeling damages on civil infrastructures after an earthquake where there are more than two hundred different labels with some of them similar to each other and the experts labeling the sets frequently disagree on which one to use. To account for this lack of consensus one solution is to get a bigger training set. If we only get images labeled by an expert the NN produced will learn to classify like this expert; but that may not be the general consensus in the field. Therefore, we need to get several experts to evaluate the same data set but these experts are not easy to get and it takes a long time to generate this sets. Another problem occurs when during inference the NN tries to classify an image with a damage that has not seen before during training. At this point the NN is guessing without providing a measure of the uncertainty. In this paper we evaluate this uncertainty using ideas from belief networks combined with a mathematical blending of similar expert damage assessments. The assessed quality of the experts together with their judgments is used to "fuzzy" the output of the NN by providing a measurement of uncertainty on the data, increasing the robustness of the automatic classifier. The output of our NN will not just be the label name but also the uncertainty associated to the label. For this paper we did test the result on synthetic generated data to prove the validity of the algorithm. Results show that, experts labeling the images have a significant influence in the accuracy for the algorithm and that the reliability of the result is significantly higher than using just one expert.

2 Introduction

NN have been hugely successful in many classification tasks, from winning the game of Go against the best human player to early skin cancer detection [15]. Still NN can easily be fooled [4] giving high confidence predictions for unrecognizable images. Traditional NN are trained to produce a point estimate by optimizing a set of tunable parameters, the optimization is typically carried out using some form of gradient descent. For example a NN can be trained with labeled images of dogs and spiders. During the inference (deployment after training) the NN will be able to automatically label new images of dogs and spiders. But what happen if during inference we feed the network the image of a cow? It will classify the image as a dog with high probability, since a NN output predictive probability is just the probability with respect to the other labels, and a dog label is more probable then a spider. The NN output predictive probabilities are often erroneously interpreted as model confidence. A NN can be uncertain in its prediction even with a high softmax output. This type of problem will happen when the assumption of having distinct classes is not met for example when out of distribution test data (like the dog, spider example), incomplete data (dog is partially hidden), trying to learn from small amounts of data , and other cases. This uncertainty is very important and in some cases it can cost lives, Tesla incident [3] where a NN did classify a white truck in front of the self-driving car with a very clear path to advance and crushed the car killing the pilot. Some NN have also been tricked about what kind of road sign they are seeing[4].

In this paper, we don't just assign a value to each of the experts for example expert A is 70 percent of the time correct while expert B is 20 percent of the time correct; and weight down everything the expert labels by that same value since that is just a rough estimate that doesn't help with accuracy. What we propose in this article is to find a way to evaluate the uncertainty for each of the labels the experts tags and use this

uncertainty estimation on the NN output. In Figure 1 we show a real example of two different experts labeling same image, it can be seen that there are clear differences and we want to use this data to estimate the quality of the expert. This way if a model returns a result with high uncertainty we can decide to pass the input to a human for classification, instead of returning a completely wrong and potentially dangerous label.

Figure 1: Two Experts Labeling Same Image

2.1 Previous work

There are several recent papers on how to capture the model uncertainty a posteriori by using Bayesian Neural Networks. [4] Uses Bayesian learning to quantify posterior uncertainty on deep NN models parameters; considering the matrix variate Gaussian to develop a scalable Bayesian outline inference algorithm by adopting a probabilistic backpropagation framework and stochastic gradient Markov Chain Monte Carlo (MCMC) on synthetic data. [5] Analyzes the different kinds of uncertainty in the model and focus its work on the importance of adding aleatoric uncertainty (can't be explained away given enough data) to the model; proposes the use of Bayesian NN for computer vision tasks improving 1-3% the model performance. [6] Analyzes NN model certainty; in the paper they prove that the dropout layer can be used as a Bayesian approximation of a well known probabilistic model, the Gaussian process. The paper uses these outputs to determine the model uncertainty and propose to pass the input to a human for classification if the output has high uncertainty. In [7] proposes the use of belief functions to represent imprecise and or uncertain knowledge of class labels (soft labels) and proposed changes to common clustering algorithms to adapt to these types of labels, presenting result on synthetic data. All the papers mentioned above try to estimate the uncertainty of the model by running same DNN with different parameters, input data and loss functions; none of them do include as an input to the model the uncertainty from the experts as we propose in this paper. In [16] a version of the segmentation algorithm SegNet that also outputs the uncertainty of the segmentation regions is presented, and is used on segmentation of street scenes. The authors provide as an autput the uncertainty on each frame for the segmentation enabling users to decide on actions if the uncertainty is high. In general Bayesian Neural Networks (BNN) do not have fixed weights for the neurons but a distribution, quantifying the uncertainty in a NN which allows to find images for which the net is unsure of their prediction, but several experiments with BNN [17] show that they also provide a high level of certainty even for out of distribution test data, and they do require long training times. concluding that a Bayesian neural network with Monte Carlo dropout is too crude of an approximation to accurately capture the uncertainty information when dealing with image data. In our paper we approach the problem in a different way than BNN and instead of adding a probability distribution to the weights of the neurons we will ask the expert labeling the images for their certainty and than through statistical analysis using belief networks [9]-[12] we "fuzzy" out the predictive output for the NN. To the best of the authors knowledge this approach has never been used before.

2.2 Structure of Paper

The remainder of this paper is organized as follows. Section 3 explains how to create a probability density function representing the certitude of an expert's assessment of a (label,image) pair. It may occur that many experts provide an evaluation of the same (label,image) pair. Section 4 explains how these multiple evaluations

may be combined into a single combined evaluation for a particular (label,image) pair. This combination represents a consensus opinion of the (label,image) certitude by averaging, as it were, the evaluations of all expert opinions. These preparations are made in support of using a belief network model (described in Section 5) to get a final quality assessment of both the NN conclusions and the quality of the expert. Finally, Sections 6 and 7 presents the simulations results, the conclusions, and future work, respectively.

3 Creating a Probability Distribution Function for Each Expert

The process begins by giving N photographs, taken in the aftermath of an earthquake, to an expert for the assessment of structural damages. During these assessments, experts will assign some number of labels to each image. These labels indicate the severity, type, and location of damage exhibited by the structure in the photograph. A truth source for these images is assumed to be available in the form of field reports or assessments from a more highly experienced expert. For our initial experiments we have a set of 30 images labeled for damages by 5 different experts.

3.1 Basic Assignment

Begin by defining the random variable V to be the conditional probability that expert E assigns label L to a given photograph Φ. In symbols

$$V = V_{L|E,\Phi} = P[\text{label } L \text{ is assigned by } E \text{ to photograph } \Phi\ ].$$

The variable V is taken to be a continuous random variable whose value is affected by the intrinsic quality of the expert, the mental state of the expert at the time of the assessment, the clarity of the photograph, and so on. A conditional probability density function for V is constructed based on two metrics:

- (1) the expert's self assessment of the likelihood of the assignment of a particular label to the given photograph (call this V^*), and
- (2) the expert's success percentage compared to the truth source (call this $\tilde{p}$).

Symbolically, the conditional density is denoted by

$$f_{L|E,\Phi}(x) \qquad 0 \le x \le 1$$

and represents a measure of the certainty (x) attached to each (label,expert,image) association. That is, the probability that the expert's assessment of label likelihood (for a given photograph) falls below x may be measured by

$$P[L \le x \,|\, E, \Phi] = \int_0^x f_{L|E,\Phi}(y)\, dy \qquad 0 \le x \le 1.$$

A common tactic ([10],[12]) is to adopt a triangularly shaped function as a measure of this density (as shown below in Figure 2). The width and position of the triangle's base expresses the range of certainty of

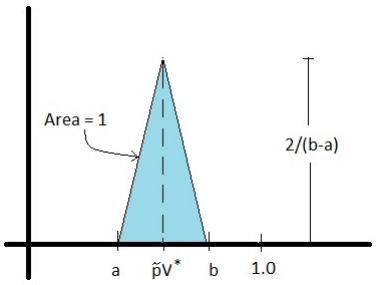

Figure 2: Density Function Shape

the $(L|E,\Phi)$ conditional association. Each expert is required, when making an $(L|\Phi)$ association, to assign a subjective estimate to the random variable V. The shape of the triangle is chosen so that the mean of the conditional density is close to (and often equals) the expert's self-assessment value, weighted by an estimate of that individual's success percentage (quality).

This centering and base width (indicated by the points a and b) of this triangular distribution is defined by placing the peak of the triangle over the value corresponding to the self-assessment, V^*, multiplied by

the expert's success percentage $\tilde{p}$. The product $(\tilde{p}\,V^*)$ is called the weighted self-assessment of the $(L|E,\Phi)$ association. The values of a and b are tied to a confidence interval placed around p (the expert's true quality). Descriptions of a, b, V^*, and $\tilde{p}$ follow.

3.2 Weighted Self-Assessment: V^* Component

As part of the assessment process, each reviewer is required to supply a self-assessment of their certitude. According to [9], a self-assessment using familiar qualifiers provides a good mechanism for self-evaluation. Since we want the schema probabilistic we use values in the range $[0, 1]$ to represent the values assigned to the qualitative descriptors as follows:

`Definitely:0.92±` `Almost Certain:0.64±` `Probably:0.36±` `Maybe:0.08±`

In this case the $\pm$ indicates the interval of ± 0.14 around the center point values $0.92, 0.64, 0.36, 0.08$ associated with the qualifiers `Definitely, Almost Certain, Probably,` and `Maybe` (respectively). That is, for example, a mark of `Definitely` suggests a certitude lying in the interval $[0.95 - 0.14, 0.95 + 0.14]$ truncated to $[0.81, 1]$ since probabilities must be between 0 and 1 (the assessment `Maybe` is also truncated from $[0.08 - 0.14, 0.08 + 0.14]$ down to $[0, 0.22]$). When an expert makes their selection of certainty for a given (L, Φ) pair, set $V^* = C$ where C is the center point value of the interval the expert selected.

To gather this data the GUI shown in figure 3 is used. As can be seen the expert who is labeling the data not only selects the region and label associated to the region but also his/her certainty concerting the label assignment.

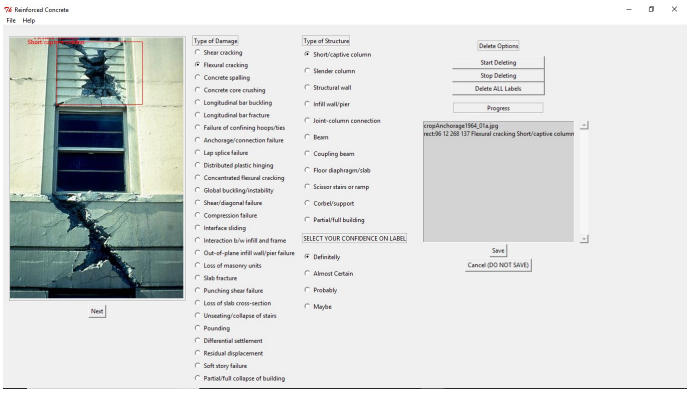

Figure 3: GUI for gathering expert certainty per label-image

3.3 Weighted Self Assessment: $\tilde{p}$ Component

The expert, E, labels a group of images and creates for each image a set of (label,image) pairs; note that each image may have one or more labels attached to it.

In conjunction with the expert's photographic evaluations described above, there are going to be field reports compiled from first-hand inspections of the damage site(s). This comprises the truth set against which the expert is evaluated. The expert is assigned a quality score that reflects both the number of agreements between the expert and the truth set and the number of disagreements – disagreements include labels identified by the truth set that were *omitted* by the expert as well as *additions* by the expert that are not part of the truth set.

For each image (call it Φ), let e_Φ, t_Φ, c_Φ represent, respectively, the total number of labels identified by the expert, the total number of labels identified by the truth set, and the total number of labels common to both the expert and the truth set. The quality rating for that expert vis-a-vis image Φ is given by the ratio of agreements to the total number of identifications (agreements plus disagreements):

$$\text{Ratio} = \frac{c_\Phi}{c_\Phi + (e_\Phi - c_\Phi) + (t_\Phi - c_\Phi)} = \frac{c_\Phi}{e_\Phi + t_\Phi - c_\Phi}.$$

Here is an example: if the expert determines photo Φ_1 should be associated with labels A, B, and C and the field reports show that photo Φ_1 contains damage captured by labels A, B, D and G, then $e_{\Phi_1} = 3$, $t_{\Phi_1} = 4$, and $c_{\Phi_1} = 2$. The expert's rating for image Φ is then

$$\text{Photo 1 rating} = \frac{c_\Phi}{e_\Phi + t_\Phi - c_\Phi} = \frac{2}{4 + 2 - 3} = 0.666\ldots$$

For the complete collection of photos evaluated by expert E, the success percentage is computed by forming the ratio of the total number of successes found over all photographs examined by the expert to the total number

of successes, omissions, and additions made by the expert. This ratio is the success proportion, $\tilde{p}$, and is given by

$$\tilde{p} = \frac{S}{M} = \frac{\sum_\Phi c_\Phi}{\sum_\Phi (e_\Phi + t_\Phi - c_\Phi)},$$

where the sums are taken over all images examined by the expert, E. In what follows, the letters S and M will be used to denote the numerator and denominator of the quantity $\tilde{p}$:

$$\begin{aligned} S &= \sum_\Phi c_\Phi \\ M &= \sum_\Phi (e_\Phi + t_\Phi - c_\Phi). \end{aligned}$$

3.4 Triangle Base Width

The proportion of successes, $\tilde{p} = S/M$, is known to be a good estimate of the probability, p, that the expert E assigns a correct label to a photo. To estimate statistically how close $\tilde{p}$ is to p, a confidence interval, $[L, R]$, is constructed around the parameter p. The natural confidence interval to be used is that for a proportion: well-known and given by the interval

$$[\, L, R \,] = [\, \tilde{p}\big(1-\epsilon\big), \tilde{p}(1+\epsilon) \,],$$

where $\epsilon = z_{\alpha/2}\sqrt{(1-\tilde{p})/(\tilde{p}M)} > 0$ (the quantity M is assumed to be large, i.e. $M \geq 30$). The confidence level, $(1-\alpha)$, is the probability to be assigned (in this analysis) to the truth of the statement: $L \leq p \leq R$. Typical choices for α are 0.1, 0.05, or 0.01 yielding 90%, 95% and 99% confidence intervals, respectively, around p. The choice of α determines the value assigned to the expression $z_{\alpha/2}$ and is easily determined from the standard normal distribution function — in the three cases cited here $z_{\alpha/2}$ is $1.644853627, 1.959963985, 2.575829304$ for $\alpha = 0.1$, 0.05, and 0.01 respectively. Once α has been chosen, and the expert's values of S and M calculated, all quantities in the equation above are known.

The base of the triangular distribution function for V is found by scaling the interval $[L, R]$ by V^*. That is the endpoints of the base of the triangle are given by $a = V^*L$ and $b = V^*R$. It is desired to keep $0 < V^*L < V^*R < 1$. Therefore, set $a = 0$ if $V^*L < 0$, set $b = 1$ if $1 < V^*R$. In summary:

$$\begin{aligned} a &= \max\ (0, V^*\tilde{p}(1-\epsilon) \\ b &= \min\ (1, V^*\tilde{p}(1+\epsilon)\ . \end{aligned}$$

The max and min functions guarantee that the base of the triangle will fall within the interval $[0, 1]$. Also note that most of the time the interval has the shape $[V^*\tilde{p}(1-\epsilon)\ ,\ V^*\tilde{p}(1+\epsilon)]$, and is

- small when $\tilde{p}$ is near 1 – the expert is good, and
- large when $\tilde{p}$ is near 0 – the expert is not good.

with $V^*\tilde{p}$ at the mid-point of the interval.

3.5 Triangle Height : Uncertainty

To make a triangle erected over the interval $[a, b]$ into a density function, the area under this triangle must equal one. When uncertainty in the expert's assessment is present, $a \neq b$ and the height of the triangle is $2/(b-a)$. Formally:

$$f_{L|E,\Phi}(x) = \begin{cases} 0 & \text{if } x < a \\ \text{linear} & \text{between } (a, 0) \text{ and } (V^*\tilde{p}, 2/(b-a)) \\ \text{linear} & \text{between } (V^*\tilde{p}, 2/(b-a)) \text{ and } (b, 0) \\ 0 & \text{if } b < x \end{cases}$$

Most of the time, this triangle will be isosceles unless $a = 0$ or $b = 1$.

3.6 Triangle Height : Certainty

When there is certainty in the knowledge of an implication or value, the corresponding certainty functions are degenerate triangles (that is, triangles with a base width of zero). In this case the density function will have this shape $f(x) = \delta(x - V^*)$ where $\delta()$ is the Dirac delta (an impulse).

3.7 Example

Here we present a brief example of how to evaluate the quality of two experts assignment of labels to two photos. The label name indicates a damage/structure type and its location, for example: "shear flexture Short column 4 100 190 300" will be label A. This indicates that different damage names with same locations will be consider different labels.

Expert 1	Expert 2	Ground Truth
Photo 1 label A Maybe: 0.08 label B Certain: 0.64 label C Maybe: 0.08 $e_\Phi = 3 \;\; c_\Phi = 2$ $M_{1,1} = e_\Phi + t_\Phi - c_\Phi = 3$ Photo 2 label D Certain:0.64 $e_\Phi = 1 \;\; c_\Phi = 1$ $M_{1,2} = 1$	Photo 1 label A Definitely: 0.92 label B Certain: 0.64 $e_\Phi = 2 \;\; c_\Phi = 2$ $M_{2,1} = e_\Phi + t_\Phi - c_\Phi = 2$ Photo 2 label D Definitely:0.92 $e_\Phi = 1 \;\; c_\Phi = 1$ $M_{2,2} = 1$	Photo 1 label A Definitely: 0.92 label B Definitely: 0.92 $t_\Phi = 2$ Photo 2 label D Definitely:0.92 $t_\Phi = 1$
$\tilde{p}_1 = 3/4$	$\tilde{p}_2 = 1$	$\tilde{p}_{\text{gt}} = 1$

Out of these values it is now possible to determine the shape of each of the triangular certainty functions. In section 3.1 the certainty functions are denoted by $f_{L|E,\Phi}(x)$. Note that for these intervals, a confidence level of 0.05 was assumed.

(L, E, Φ)	a = **L**eft	V$^*\tilde{p}$ = **C***enter*	b = **R**ight
$(A, 1, 1)$	0.026	0.06	0.094
$(B, 1, 1)$	0.208	0.48	0.752
$(C, 1, 1)$	0.026	0.06	0.094
$(D, 1, 2)$	0.208	0.48	0.752
$(A, 2, 1)$	0.92	0.92	0.92
$(B, 2, 1)$	0.64	0.64	0.64
$(D, 2, 2)$	0.92	0.92	0.92

The certainty functions obtained from the values for photo 1 (stated above) are represented in Figure 4. It can be seen that the certainty functions associated with expert 1 (in shades of blue[1]) are triangular while the certainty functions associated with expert 2 are impulsive (in shades of green). This is due to the fact that expert 1 made several mistakes in assigning labels (compared to the truth set) while expert 2 is not only certain about the labels but has also made correct assignments[2].

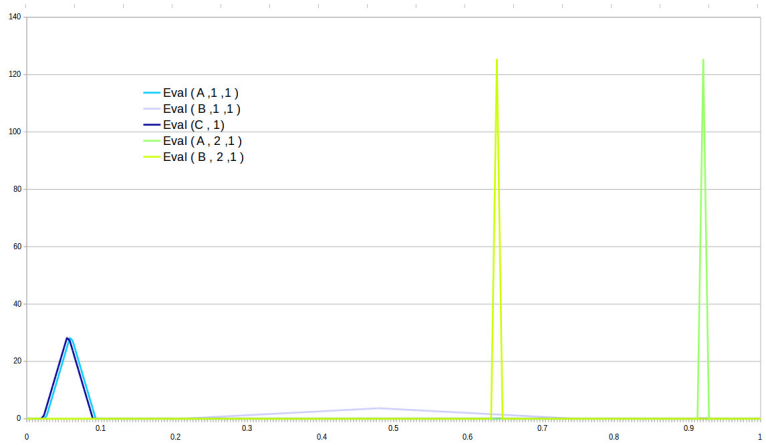

Figure 4: Experts Density Function Shapes (Photo 1)

[1]The certainty functions associated with $(A, 1, 1)$ and $(C, 1, 1)$ are identical. For clarity, the graph associated with $(C, 1, 1)$ has been shifted a small amount to the right.

[2]At this stage of development, a triangular density function with a very narrow base has been used instead of an impulse

4 Combining Multiple (label,image) Expert Evaluations

To model a collection of photos and labels as a belief network, each (label,image) pair is treated as an edge in a directed graph (see Section 5). To simplify these directed graphs, multiple evaluations of same (label,image) pair by different experts are combined into a single composite evaluation; this results in a network with at most one directed edge between photos and labels; figure 5 shows a graph of this process.

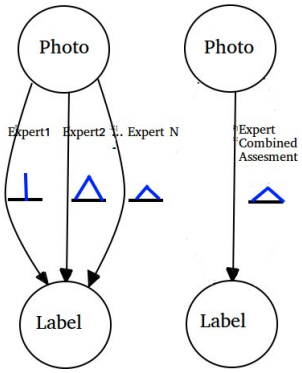

Figure 5: Combine Experts Evaluations into One Edge

Assume that the image, Φ_s, has been evaluated independently by K experts each of whom have assigned label L_j to the Φ_s. Assume further that the assignment of images to experts has been done in an unbiased fashion. That is, for example, an expert on structural damage near windows is just as likely to receive a window image as a non-expert (simple modifications in this analysis can be made to accommodate a biased distribution of images).

These assumptions yield the following relationship

$$V_{L|\Phi} = P[L_j \,|\, \Phi_s] = \frac{1}{K}\sum_i P[L_j \,|\, E_i, \Phi_s] = \frac{1}{K}\sum_i V_{L|E_i,\Phi}.$$

That is, the certitude function for $V_{L|\Phi}$ is the certitude function for the arithmetic average of the variables $V_{L|E_i,\Phi}$. Since the variables $V_{L|E_i,\Phi}$ are assumed to be independent, elementary probability theory provides a way to computing the certitude function for $V_{L|\Phi}$ using two basic equations:

$$\begin{aligned} f_{aX}(x) &= \frac{1}{a} f_X(\frac{x}{a}) \\ f_Y(y) &= \big(f_{X_1} \star f_{X_2} \star \ldots \star f_{X_K}\big)(y), \end{aligned}$$

where a is a constant, $Y = \sum_i X_i$ is the sum of K independent random variables, and $\star$ represents the convolution operation

$$(f \star g)(x) = \int_{-\infty}^{\infty} f(t)g(x-t)\,dt.$$

In this application, $a = 1/K$ and $X_i = aV_{L|E_i,\Phi}$.

4.1 Example

Continuing with the example in section 3.7, the expert's evaluations yield (label,image) pairs with the following values:

```
(L,E,Φ): label,image pair (L,Φ), Expert E, triangle base:    a;   V*p̃;    b
=========================================================================
(A,1,1): label,image pair (A,1), Expert 1, triangle base:  0.026; 0.06; 0.094
(A,2,1): label,image pair (A,1), Expert 2, triangle base:  0.920; 0.92; 0.920
(B,1,1): label,image pair (B,1), Expert 1, triangle base:  0.208; 0.48; 0.752
(B,2,1): label,image pair (B,1), Expert 2, triangle base:  0.640; 0.64; 0.640
(C,1,1): label,image pair (C,1), Expert 1, triangle base:  0.026; 0.06; 0.094
(D,1,2): label,image pair (D,2), Expert 1, triangle base:  0.208; 0.48; 0.752
(D,2,2): label,image pair (D,2), Expert 2, triangle base:  0.920; 0.92; 0.920
```

Notice that (label,image) pairs $(A, 1)$, $(B, 1)$, and $(D, 2)$ have each been evaluated by experts 1 and 2. The certitude functions for the two evaluations of $(A, 1)$ are convolved together to form a combined certitude function. Note that each expert's certitude function for this (label,image) pair is weighted equally in the convolution.

Figure 6 (left) shows the combined certitude function result for pair $(A, 1)$ as well as the two input certitude functions $(A, 1, 1)$ and $(A, 2, 1)$. The input certitude functions are in blue and red with the combined certitude output represented in purple. The figure 6 (right) shows the same action applied to combine the two evaluations for label,image pair $(B, 1)$. On the left image, the combined certitude is representative of an average of the two input certitudes – both the expert's evaluations, (as represented by the center location of the triangle base) and quality of the expert (represented by the width of the triangle base) appear to have been averaged. For the chart on the right, expert 1 has a flattened certitude function with a peak over 0.48 while expert 2 is asserting certainty about the evaluation 0.64. Note in this case that the combined certitude has its peak over the value 0.56 (the average of the two input certitude peaks) but has lost the sharpness of expert 2's evaluation while improving the dullness of expert 1's evaluation. Similar certitude functions may be constructed for labeled C and D.

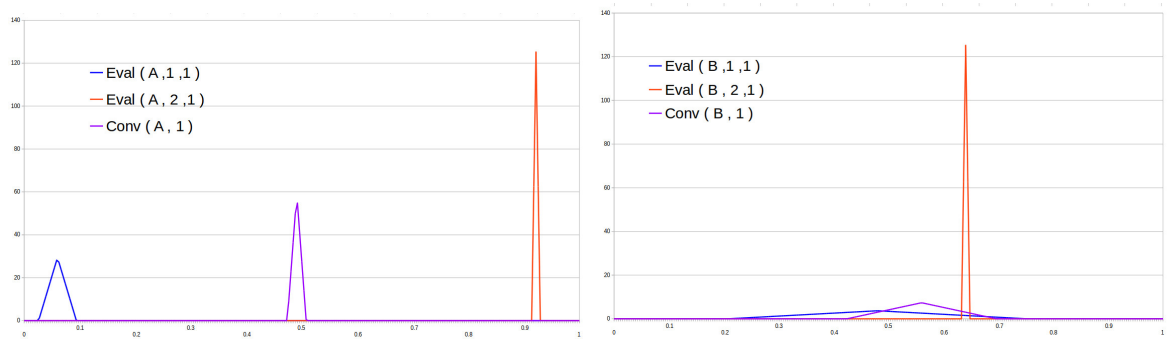

Figure 6: Combined Experts 1 and 2: (Label A:Photo 1)-Left; (Label B:Photo 1)-Right

5 Belief Network for Assessing The Uncertainty of the Label

A Belief Network (BN) is a directed graph where each node represents an event, an object, or some similar static item and the directed edges indicate the presence of causality or dependence (see [11]). To model a collection of images and labels as a BN, the nodes comprise the collection of images and labels while the directed edges correspond to the (label,image) pairs. Normally a conditional probability (a number) is attached to each node and each directed edge in this graph (by either observation, analysis, or computation) to represent the likelihood of a node (an event) or edge (a causality) occurring. In this work, the conditional probabilities associated with the edges are replaced by the certitude functions associated with each (label,image) pair. Nodes at the receiving end of each directed edge then also inherit a certitude function rather than a single value.

Consider the belief network shown in figure 7. The nodes L_j represent a list of three labels, the nodes Φ_i represent three photographs in the image collection, and Φ represents a new (unseen by any of the experts) image. The directed arrows travelling from the image collection, $\{\Phi_i\}$, to the set of labels, $\{L_j\}$, represent the evaluations done by the experts and the directed arrows going from the given, new image, Φ, to the image collection describe the amount of similarity between the new image and the images in the image collection. Note that the photos Φ_1, Φ_2, Φ_3 are the same images that have been used during the training of the NN.

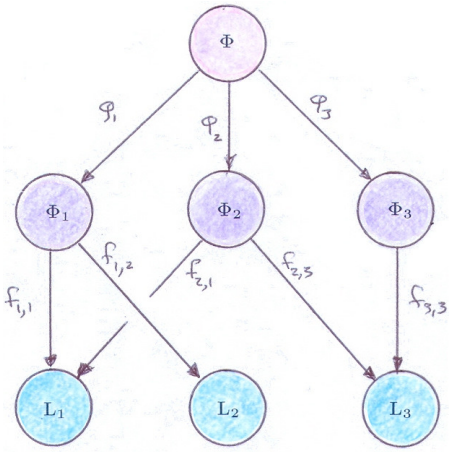

Figure 7: Example of Belief Network

Suppose it is determined (somehow) that Φ contains an area with a 60% similarity with an area on photo 1 (Φ_1), a 60% similarity with and area on photo 2 (Φ_2), and a 30% similarity with an area on photo 3 (Φ_3). A probabilistic restatement would amount to asserting that the similarity of Φ with part of photo 1 is the same as Φ's similarity with part of photo 2 but the similarity of Φ with photo 3 is only half as much. Translating this into probabilities yields value of 6/15, 6/15, and 3/15 respectively for the probability that, given Φ, the image Φ_i resembles Φ. In this example, each of these conditional probabilities is computed by dividing the original belief similarity by the sum of all belief similarities. Thus, each link, $\Phi \to \Phi_i$, in the top row of this BN receives a probabilistic similarity weighting, $\phi_i = P[\Phi_i \,|\, \Phi]$. Note that the relationship among the ϕ_i is $\sum_i \phi_i = 1$.

The probabilities residing at the photo nodes Φ_i are then given by $P[\Phi_i] = P[\Phi_i \,|\, \Phi]P[\Phi] = \phi_i P[\Phi]$. Emanating from the photo nodes are implication arrows pointing at the label nodes. Each arrow carries a conditional probability given by the functions $f_{1,1}, f_{1,2}, f_{2,1}, f_{2,3}, f_{3,3}$: the (combined where necessary) expert evaluations calculated in section 4. Therefore, each arrow holds a conditional certainty function created from the experts evaluations that links the photo node in a probabilistic way to the label node.

A BN (such as the one depicted in figure 7) provides a simple way of assigning probabilities and certainty distributions to each of the labels. For a single label, L_j, this equation holds:

$$P[L_j] = \sum_{\Phi_i} P[L_j \,|\, \Phi_i]\phi_i P[\Phi].$$

Single numerical values for the labels L_1, L_2, and L_3 are the outputs from a neural network softmax layer. That is, for example, an NN output of $L_1 = 0.7, L_2 = 0.26$, and $L_3 = 0.04$ means that the NN did recognize the existence of label L_1 in the image Φ. However, the NN does not supply probability values for the photos ($P[\Phi_i]$). This may be overcome because the relationships between the values $\phi_i = P[\Phi_i] (i = 1, 2, \ldots, I)$ and $l_j = P[L_j] (j = 1, 2, \ldots, J)$ are linear:

$$\mathbf{L} = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_J \end{bmatrix} = \begin{bmatrix} f_{11} & f_{11} & \ldots & f_{1I} \\ f_{21} & f_{21} & \ldots & f_{2I} \\ \vdots & \vdots & \vdots & \vdots \\ f_{J1} & f_{J1} & \ldots & f_{JI} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_I \end{bmatrix} = \mathbf{F\Phi},$$

where $f_{ij} = P[L_j \,|\, \Phi_i]$. Notice that the conditional quantities on the right hand side of this equation have known conditional distribution functions and from this, the distributions of the quantities in the L vector may be determined provided that the values of $\phi_i = P[\Phi_i]$ are known.

In the illustration at hand (figure 7) assume that the links from photo Φ_i to labels L_j have the certitude functions described in figure 6 above[3] and the probabilistic values for the numbers ϕ_i are $6/15, 6/15$, and $3/15$. Here is a table showing the association to be used in this illustration. Figure 8 shows the label certitude functions that arise from these image data probabilities for the BN described above.

[3]In the previous example there were two photographs and two labels. The certitude functions developed there will be reused here with three photographs and three labels as illustrated in the table to follow.

$$
\begin{array}{lcl}
\Phi_1 \rightarrow L_1 & \Leftrightarrow & (A,1) \\
\Phi_1 \rightarrow L_2 & \Leftrightarrow & (B,1) \\
\Phi_2 \rightarrow L_1 & \Leftrightarrow & (D,2,2) \\
\Phi_2 \rightarrow L_3 & \Leftrightarrow & (C,1) \\
\Phi_3 \rightarrow L_3 & \Leftrightarrow & (D,1,2)
\end{array}
$$

$$\phi_1 = 6/15 \quad \phi_2 = 6/15 \quad \phi_3 = 3/15$$

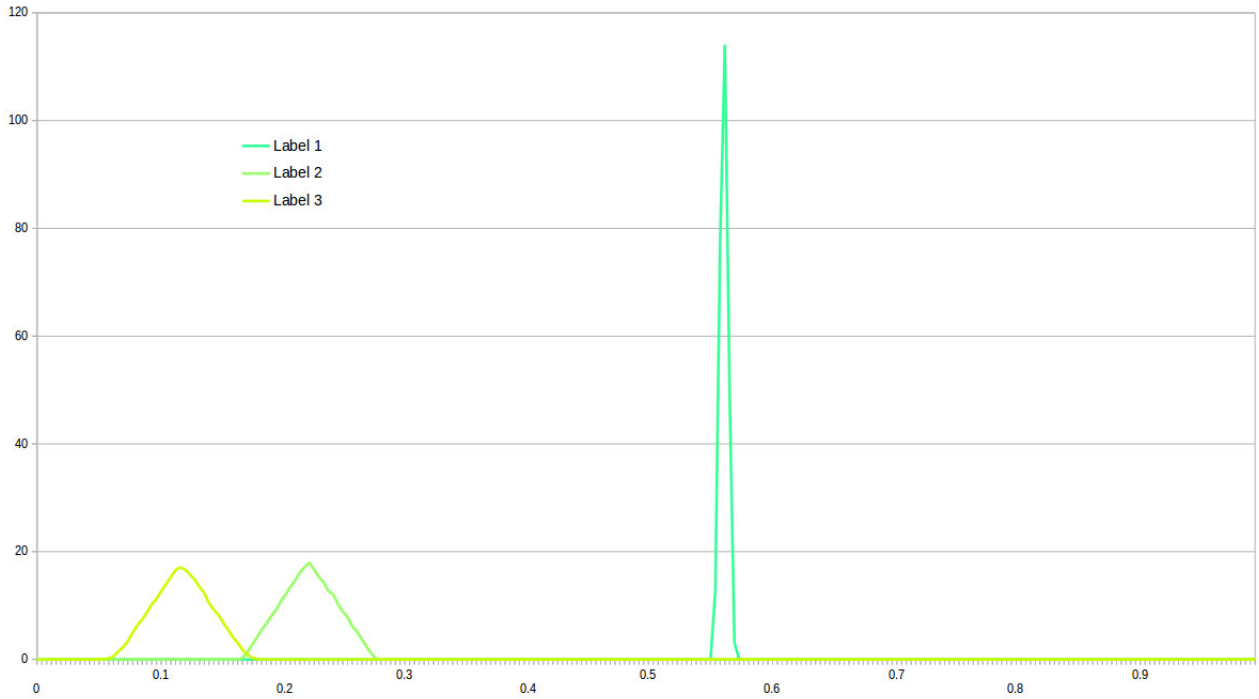

Figure 8: BN Generated Certitude Functions (from image probabilities)

The algorithm used to create the figure **??** certitude functions from the images involves performing *a priori* calculations (described below in Algorithm 1). However, if the starting point is the NN's estimation of the label probabilities, the generation of certitude functions is accomplished by inverting the linear system described above to produce candidate values for ϕ_i. The *a priori* calculations of Algorithm 1 are then performed. By using the BN in this manner, we are assigning a certainty function to each value produced by the NN (as illustrated by the red lines in figure 9).

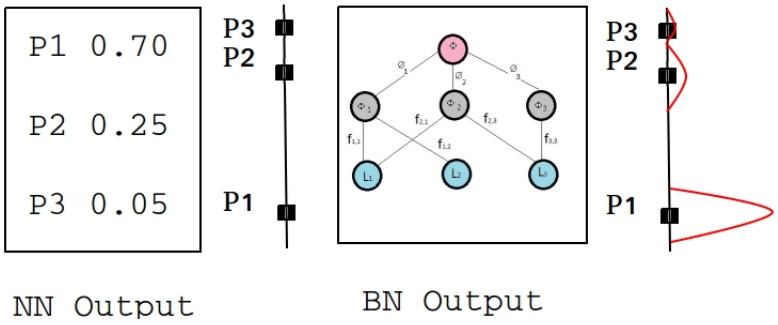

Figure 9: Example of Output of a Neural Network augmented by Belief Network Certitude Functions.

5.1 A Priori Evaluations

If the probabilities, ϕ_i, of each link $\Phi \rightarrow \Phi_i$ are known then a simulation experiment may be conducted to produce an estimate of the certitude functions of each label L_j. To perform an *a priori* evaluation of the L_j certitude functions assume that the photograph Φ has been given for damage assessment. It will be assumed

that $P[\Phi] = 1$ since all image certitude functions and all label certitude functions have the form $h(\cdot)P[\Phi]$ for some function $h(\cdot)$, independent of $P[\Phi]$. If and when probabilistic information about Φ itself becomes available (that is when $P[\Phi]$ is known), all results given below may be adjusted simply by multiplying each result by $P[\Phi]$.

To produce an estimate of the certitude functions for the labels under the assumption of constant certitude functions for the images the following algorithm is employed to construct a raw histogram. Note that $f_{j,i}$ is the certitude function for the link $\Phi_i \to L_j$ and $L_j = \sum_i f_{j,i}\phi_i$.

Algorithm 1

For each label L_j repeat:
    Set $S_j \leftarrow 0$.
    For each i in the sum:
        Transform $f_{j,i}$ into a distribution function, $F_{j,i}$ for the link $\Phi_i \to L_j$.
        Select a pseudo-random value, u, from a continuous uniform distribution over $[0, 1)$.
        $v_i \leftarrow F_{j,i}^{-1}(u)$
        $S_j \leftarrow S_j + v_i\phi_i$
    End loop
End repeat

The quantity S_j is an observation of the random process defined by $L_j = \sum_i f_{j,i}\phi_i$. Repeat this basic algorithm M times building up a sample of M observations, $S_1, S_2, \ldots, S_M$ from which a histogram, approximately proportional to the density function for label L_j, may be constructed. Scale the ordinates of this raw histogram so that area under this curve is approximately one. This normalized histogram is an approximation to the label L_j's certitude function. Repeat this task for each label. The certitude functions shown in figure 8 above were calculated following this procedure.

5.2 A Posteriori Evaluations

When the neural network processes a photograph it is going to produce values for the vector $\mathbf{L}$. The task here is then to work backwards and reconstruct a feasible vector $\mathbf{\Phi} = (\Phi_1, \Phi_2, \ldots, \Phi_I,)$ of image probabilities that then can be propagated forward through the belief network to produce a set of certainty functions for the vector $\mathbf{L} = (L_1, L_2, \ldots, L_J,)$ of labels. Note that not all photographs in the image collection need to be included, only those that bear on the labels identified by the NN as having a non-zero likelihood.

Because the number of photographs is much larger than the number of labels ($I >> J$) it is guaranteed (barring inconsistencies in the equations) that there will be infinitely many vectors $\mathbf{\Phi}$ that satisfy the linear system for a given vector $\mathbf{L}$. The initial task is to find at least one solution that also satisfies $\sum_i \phi_i \leq 1$ with $0 \leq \phi_i \leq 1$. The sum of the ϕ_i is permitted to be strictly smaller than one to allow for the fact that the image collection is likely incomplete. This condition is modeled by postulating the existence of an image Φ_{Other} to hold the remaining probability. As there are no edges connecting Φ_{Other} to any labels, this image may be ignored in the computations.

The problem of inverting the linear system $\mathbf{L} = \mathbf{F\Phi}$ may be solved using techniques from the theory of Linear Programming. To accomplish this recovery of a suitable vector $\mathbf{\Phi}$, the $\mathbf{F}$ matrix is filled with representative values from each certainty function associated with the expert's (label,image) evaluations. In the analysis presented here, $f_{i,j}$ is set equal to the expected value of the certitude function of the $\Phi_i \to L_j$ link. Here is a formal specification of the linear programming problem:

Given $\mathbf{L}$ from the NN, find a vector $\mathbf{\Phi}$ subject to these constraints

$$\mathbf{L} = \mathbf{F\Phi}$$

$$0 \leq \phi_1 \leq 1$$

$$0 \leq \phi_2 \leq 1$$

$$\vdots$$

$$0 \leq \phi_I \leq 1$$

$$\sum_{1 \leq i \leq I} \phi_i \leq 1$$

Note that no objective function is given for this linear program since the point of the computation is to identify the set of feasible vectors $\mathbf{\Phi}$. Two key facts about the solutions to this linear program are known:

- the solution space forms a convex polytope (a bounded n-dimension solid whose faces are $(n-1)$-dimensional hyperplanes), and
- every point in this polytope is a finite convex combination of the vertices of the polytope.

This implies that all solutions to the linear program have the form $\mathbf{\Phi} = \sum_j \lambda_j \mathbf{\Psi_j}$ where $\mathbf{\Psi_j}$ are the vertices of the convex polytope identified by the linear program and λ_j are positive constants that sum to one. Construction of the certitude functions now proceeds like an *a priori* evaluation:

Algorithm 2

Determine $\mathbf{\Psi}$. Choose a large integer N.
Repeat N number of times:
  Select the values of λ_j at random subject to the constraints.
  Form a solution $\mathbf{\Phi} = \sum_j \lambda_j \mathbf{\Psi_j}$
  Perform Algorithm 1 with $\mathbf{\Phi}$ and $M = 1$
  Use the output S_j from Algorithm 1 to build up a raw histogram
End Repeat
Normalize the resulting raw histogram to unit area.

6 Results

In this section, test results are shown that demonstrate that the methods described in Sections 3 through 5 can be used to assign a certainty range to the labels with which different experts tag the images. To shorten the length of the paper we used synthetic data throughout to show that the proposed method works well without having to add all the details of a real NN configuration. These results are shown in Section 3.3 and in Figures 4 and 6.

The BN described by Figure 7 has been simulated using the *a posteriori* evaluation described by Algorithm 2. The table of image,label certitude functions defined in Sections 7 and 4.1 is summarized below in Figure 10:

						(L , E , Phi)	Individual	(L , Phi)	Convolved
V*	p~	epsilon	a	V*~p	b	Designation	Mean	Convolved	Mean
0.08	0.75	0.565792867	0.026052428	0.06	0.093947572	A11	0.06	A1=A11 * A21	0.49
0.64	0.75	0.565792867	0.208419424	0.48	0.751580576	B11	0.48	B1=B11 * B21	0.56
0.08	0.75	0.565792867	0.026052428	0.06	0.093947572	C11	0.06		
0.64	0.75	0.565792867	0.208419424	0.48	0.751580576	D12	0.48		
0.92	1	0	0.92	0.92	0.92	A21	0.92		
0.64	1	0	0.64	0.64	0.64	B21	0.64		
0.92	1	0	0.92	0.92	0.92	D22	0.92		

Figure 10: Belief Network Input Data (95% confidence)

The label probabilities shown in Figure 9 ($P1 = P[L1] = 0.70$, $P2 = P[L2] = 0.26$, and $P3 = P[L3] = 0.04$) are the outputs from the softmax layer in the NN. These probabilities indicate that the NN recognizes labels $L1$, $L2$, and $L3$ as possibly present in the new input image — $L1$ being the most likely. Assuming the Belief Network illustrated in Figure 7 together with the certitude functions summarized above (Figure **??**) and an application of Algorithm 2, produces the certitude functions for the labels shown here (Figure **??**).

This graph shows the certitude functions corresponding to the three softmax layer evaluations by the NN. With respect to label $L1$, the narrow width of the base of the triangular function indicates that the softmax layer value of 0.70 is supported by the expert's opinions. The base of this triangle is the interval $[0.688, 0.711]$ with the peak over the certitude value 0.699 (all values are approximate). In probabilistic terms the probability that an expert would assign label $L1$ to the new image lies between 0.688 and 0.711 (relative error of about $\pm 1.6\%$). For labels $L2$ and $L3$ the base intervals are $[0.195, 0.342]$ and $[0.018, 0.059]$ with peaks over 0.258 and 0.039 respectively. Similar probabilistic statements apply to labels $L2$ and $L3$ (relative errors of $\pm 28.5\%$ and $\pm 52.6\%$ respectively).

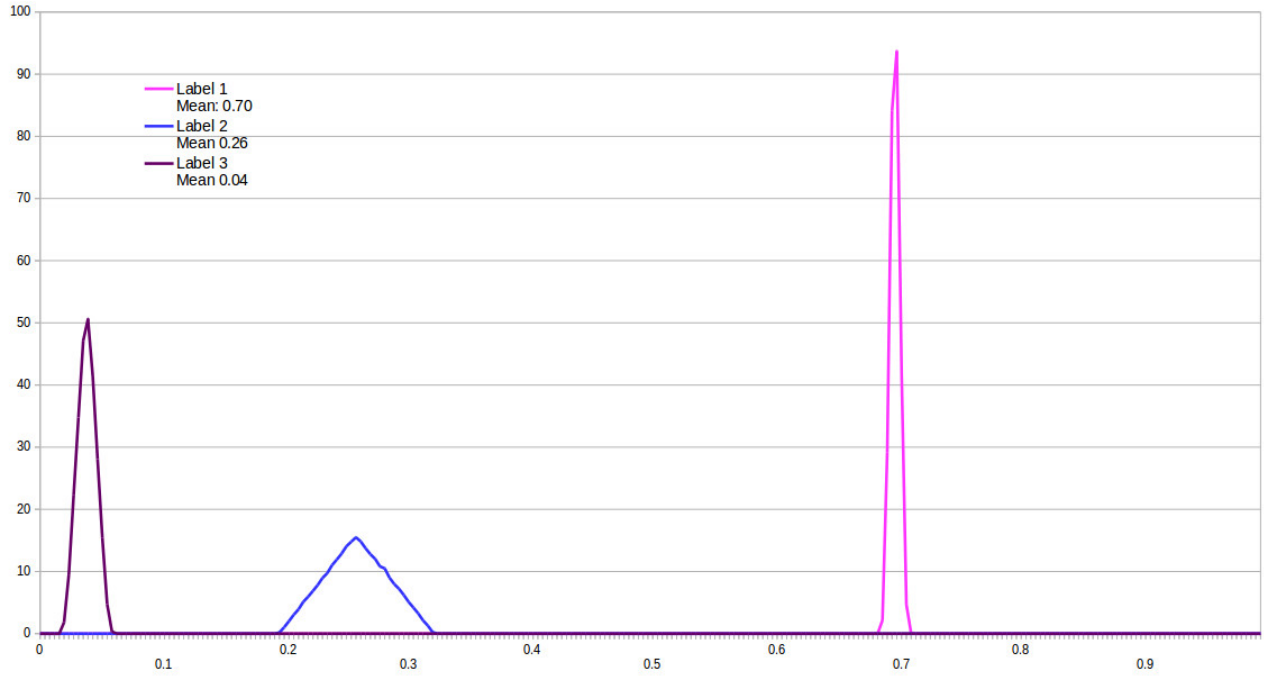

Figure 11: BN Generated Certitude Functions (from label probabilities)

7 Summary, Conclusions, and Future Work

At this stage, it appears that BNs do provide a valid estimate for the certainty of the label and can be used to "fuzzy" out the output of a NN for object classification. For NNs to be come widely accepted in commercial applications the algorithms needs to provide some measure of the uncertainty to avoid catastrophic and embarrassing failures of the classifier. (If the reader is interested in replicating the experiments and calculating the BN, you can find all code use here in the github [19].)

In this project we developed a method to evaluate the quality of different experts manually tagging pictures used for training of Neural Networks. We use probabilistic theory based on Belief Networks to evaluate the work of experts that allows us to gather information about how consistently experts are tagging the images and with what certainty they do so. We provide also a method based in probability theory to combine different experts tagging of the same images to obtain a more accurate set of training images that can later be used in training a NN for automatic classification. Finally, the assessed quality of the experts and their evaluations can also be used to "fuzzy" the output of the Neural Network increasing the robustness of the classifier.

Several issues/observations have been uncovered during the development of the BN. One issue is that the linear system, $\mathbf{L} = \mathbf{F}\mathbf{\Phi}$ identified in section 5 may fail to have solutions if the number of images is smaller than or equal to the number of labels (this may happen if some labels possess too few antecedent images). Options for grappling with this situation are under evaluation since it is desired to use no more labels than the number identified by the NN softmax layer as having a non-zero probability. Note that the example shown in this paper illustrates this issue – there are three images and three labels. With this example, slight inconsistencies between the NN data and the BN will cause this issue to appear.

A second observation arises in the realization that the BN can be used to improve the quality evaluations given to each expert. In this paper, a rudimentary method was used to assess this quality (see section 3.3). It seems likely that combining the BN with the NN's truth set can provide a way for the refinement of each expert's quality factor, $\tilde{p}$. Further study of these two issues/observations is underway.

The general method proposed in this paper can be used with any training set where the labels (1) contain uncertainty and (2) were provided by different experts. Currently we have an actual set of earthquake damage (label,image) evaluations done by several different civil engineering experts. Preliminary work is showing encouraging results. Detailed descriptions of this specific NN will be presented in a future paper.

References

[1] Patterson B, Leone G, Pantoja M, Behrouzi A. Deep learning for automated image classification of seismic damage to built infrastructure Proceedings of the 11th National Conference in Earthquake Engineering 2018

[2] Pantoja M, Fabris D., Behrouzi A. "Deep Learning Basic Overview" Concrete International Magazine September 2018

[3] Tesla Crash Preliminary Report US department of transportation NHTSA PE 16-007

[4] Sun S, Chen C, and Carin L Learning Structured Weight Uncertainty in Bayesian Neural Networks Proceedings of the 20th International Conference on Artifi- cial Intelligence and Statistics (AISTATS) 2017 Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54

[5] Kendall A and Gal Y What uncertainties do we need in Bayesian Deep Learning for Computer Vision NIPS 2017 https://arxiv.org/abs/1703.04977

[6] Gal Y, and Ghahramani Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning Proceedings of The 33rd International Conference on Machine Learning PMLR 48:1050-1059, 2016

[7] Deceus T. Handling imprecise and uncertain class labels in classification and clustering .Bayesian Deep Learning COST Action IC 0702 Working group C, Mallorca, March 16 2009

[8] What my Deep Learning model Doesn't know Y Gal

[9] `www-compsci.swan.ac.uk\csphil\CS345\chapts5-9.pdf`

[10] Hackerman, David The Certainty-Factor Model, Encyclopedia of Artificial Intelligence Second Edition Wiley, New York pp. 131-138

[11] Pearl, Judea Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference Morgan Kaufmann San Mateo CA

[12] Klir, George J.(editor), Yuan, Bo(editor) Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems –Selected Papers by Lotfi A. Zadeh, Advances in Fuzzy Systems – Applications and Theory Vol 6 World Scientific

[13] Knuth, D. E. The Art of Computer Programming, Vol 2, Section 4.3.3, pp 290-295

[14] Press, W. H., et. al., Numerical Recipes in C, Section 8.10, pp 329-343.

[15] Google Research Research Blog: AlphaGo: Mastering the ancient game of Go with Machine Learning. 27 January 2016

[16] Alex Kendall and Vijay Badrinarayanan and Roberto Cipolla, Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding, CoRR, abs/1511.02680, 2015, http://arxiv.org/abs/1511.02680, arXiv, dblp computer science bibliography, https://dblp.org

[17] Hendrik J. Weideman Quantifying Uncertainty in Neural Networks https://hjweide.github.io/quantifying-uncertainty-in-neural-networks

[18] Avis, David and Fukuda, Komei A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra Discrete & Computational Geometry 1992, Sep,01 v. 8,n. 3,pp.295–313

[19] GitHub address to place all the code