Relaxed Softmax: 
Efficient Confidence Auto-Calibration for 
Safe Pedestrian Detection

Lukas Neumann  Andrea Vedaldi  Andrew Zisserman
Department of Engineering Science
University of Oxford
{lukas,vedaldi,az}@robots.ox.ac.uk

Abstract
As machine learning moves from the lab into the real world, reliability is often of paramount
importance. The clearest example are safety-critical applications such as pedestrian detection
in autonomous driving. Since algorithms can never be expected to be perfect in all cases,
managing reliability becomes crucial. To this end, in this paper we investigate the problem
of learning in an end-to-end manner object detectors that are accurate while providing
an unbiased estimate of the reliability of their own predictions. We do so by proposing
a modification of the standard softmax layer where a probabilistic confidence score is
explicitly pre-multiplied into the incoming activations to modulate confidence. We adopt
a rigorous assessment protocol based on reliability diagrams to evaluate the quality of the
resulting calibration and show excellent results in pedestrian detection on two challenging
public benchmarks.

1 Introduction
Deep neural networks have been very successful at addressing many real-world classification, detection and
recognition problems. As a result, they have already moved from being a subject of research to being deployed
“in the wild”, where their decisions impact everyday life.
Among such applications, some, such as pedestrian detection in autonomous driving, are critical for safety. In
such cases, the ability of networks to make reliable decisions is of paramount importance. As new architectures
and components are introduced, research seems to be focused on improving accuracy and speed of networks
on average. Yet, far less attention has been given to determining when predictions can be trusted and when
they cannot. In other words, neural networks are good at providing answers which are correct most (but not
all) of the times, but not good at telling when such answers can be trusted, or when, instead, they amount to
little more than an educated guess.
 Having a measure of the network’s prediction reliability is essential in many applications. While the strength
of activations in deep neural networks is often correlated with the network’s confidence, such values can
hardly be mapped in a systematic manner to a meaningful and verifiable measure of confidence. Consider an
application such as pedestrian detection using a camera mounted on the front of a car. In most situations, the
visual information is clear and the network can determine with confidence whether a pedestrian is contained in
the field of view of the camera or not. However, in some cases viewing conditions are much poorer. A good
example is pedestrian detection at night (Figure 1), in which poor, unusual, and rapidly changing illumination,
shadows, reduced color sensitivity, and tendency of colors to fuse in the background make reliable detection
exponentially harder.
In such cases, we would like to know when the network is not able to make an accurate determination of the
possible presence of pedestrians in front of the car, for example in order to take precautionary action such
as slowing down, or relying more on other sensors. Instead, as we confirm empirically in this paper, neural
networks do not hedge their bets, and vote with exceeding confidence for one case (pedestrian present) or the
other (not present) even when the evidence in the data is genuinely scarce or ambiguous. Our first contribution
is, in fact, to assess state-of-the-art pedestrian detectors in terms of the quality of their confidence estimates
using reliability diagrams and the calibration error (CE) metrics.
Our second contribution is of a technical nature and amounts to a simple but effective modification of the
standard softmax layer used in neural networks that significantly improves its self-calibration properties. This
layer, which we call relaxed softmax, has nearly the same complexity as softmax, requiring the network to

2018 NIPS Workshop on Machine Learning for Intelligent Transportation Systems.
Figure 1: **Decision confidence in safety-critical applications (NightOwls data).** Top: detection probability estimated by a state-of-the-art neural network detector applied to pedestrians. The probabilities (number in red) do not reflect well the amount of evidence, with overconfident detections in ambiguous cases (e.g., the statue in 2nd column above). Bottom: relaxed softmax (bottom row) provides calibrated output that significantly better reflects the detection ambiguity.

estimate just one more scalar output per sample, which is of negligible computational impact in all practical architectures.

We train relaxed softmax using the standard cross-entropy loss that is already used with vanilla softmax, which should result in both being properly calibrated. However, as found by other authors [24, 8] and as we confirm in the experiments, reliable calibration does not occur for softmax. There are two possible opposite explanations: overfitting, as the model becomes overly confident on the training data, and underfitting, as the architecture may be unable to easily represent confidence.

We show empirically that both effects are present. Overfitting is present because we can improve calibration by simply rescaling all scores by a constant multiplicative factor tuned on a validation set. Underfitting is present as well, as relaxed softmax, which is used as a drop-in replacement for softmax, does significantly improve the calibration profile of the emitted probabilities and achieves a significant reduction in CE. While we show this in the context of object detection, and safety-critical pedestrian detection in particular, we expect similar benefit to apply anywhere softmax is used.

The rest of the paper is organized as follows. Section 2 summarizes the work most related to ours, Section 3 discusses our method in detail, Section 4 assess it empirically in challenging pedestrian detection scenarios, and Section 5 summarizes our findings.

2 Related Work


Recently, Guo et al. [9] studied the problem of deep networks calibration, but only focused on parametric and non-parametric calibration of pre-trained models that have no notion of uncertainty in their own right. Our model, in contrast, is specifically trained to infer its own uncertainty. Additionally, they only focus on simple classification problems based on the expected calibration error metric, which is not suitable for detection problems (see Section 3.3).

**Pedestrian Detection.** Pedestrian detection is one of the most intensely studied subfield in object detection, because of its importance in real-world applications. All current state-of-the-art methods are derived from the Faster R-CNN detector of Ren et al. [22], but they differ in the way they address the issue of detecting small pedestrians. ?] and Cai et al. [3] use a specialized multi-scale networks, whereas other methods [28, 31, 2] exploit a separate classification network which operates on higher-resolution features or directly on the source image.
The current state-of-the-art pedestrian detection method of Brazil et al. [2] also falls within this category, as their method is a two-stage pedestrian detector combined with an attention-based mechanism coming from a coarse semantic segmentation branch. In the first stage, region proposals are generated by a Region Proposal Network (RPN), and in the second stage, the proposals are classified as pedestrian or a background using a separate network that operates directly on image crops.

3 Method

This section describes our approach for end-to-end calibration of object detectors. We start by reviewing the general architecture of modern detectors based on convolutional neural networks (Section 3.1) and then introduce a relaxed softmax layer that can significantly improve the quality of the detection confidence estimated by the neural network (Section 3.2).

3.1 Background: Object Detectors

Most modern detectors, including variants specialized for the detection of pedestrians [28, 30, 31, 2], are based on the propose & verify paradigm. Namely, given an image $I$, a mechanism such as a specialized neural network [22], an ad-hoc algorithm [26], or even simply the use of a fixed pool [16], is used to generate a shortlist of candidate image regions $R_i$, $i = 1, \ldots, N$ that are likely to contain occurrences of the objects of interest. Then a neural network $\Phi(R_i; I)$ is tasked with mapping each region to a vector $(z_{i,1}, \ldots, z_{i,K+1}) \in \mathbb{R}^{K+1}$ of scores, one for each possible class identity $1, \ldots, K$ plus class $K+1$ for background. Scores are converted into probabilities by using the softmax operator $\sigma$, resulting in a normalized belief value $\sigma_i(j)$ for each hypothesis $j = 1, \ldots, K + 1$:

$$\sigma_i(j; z_i) = \frac{\exp(z_{i,j})}{\sum_{k=1}^{K+1} \exp(z_{i,k})}.$$  \hspace{1cm} (1)

In this manner, the detection problem is reduced to a standard classification problem. As commonly done in deep networks, after converting the scores produced by $\Phi$ into probabilities by using the softmax operator, scores are fitted to the ground truth label $y_i \in \{1, \ldots, K + 1\}$ of each region by minimizing the cross-entropy loss:

$$L_{ch}(z_i, y_i) = - \sum \log \sigma_i(y_i; z_i).$$ \hspace{1cm} (2)

The ground-truth label $y_i$ of each candidate region $R_i$ is determined to be positive or negative based on the Intersection-over-Union (IoU) of $R_i$ with the ground truth object annotations, which must therefore be available. This process, which effectively transfers labels from the annotated regions to the candidate regions, is required because none of the candidates will in general match exactly the ground-truth annotations.

After training, given a new image, the neural network $\Phi$ can associate to each candidate region $R_i$ a class $y_i$ with confidence $s_i$ as the maximum of the score vector $\sigma_i(j)$, $j = 1, \ldots, K + 1$:

$$y_i = \arg\max_j \sigma_i(j; z), \quad s_i = \sigma_i(y_i; z).$$ \hspace{1cm} (3)

Each candidate box is then associated with its position, size and a score, which determines how likely the candidate box is to contain the object of interest (pedestrian in our application) or not. During inference, only the boxes with the highest score are kept, using non-maxima suppression to filter redundant detections, and candidate boxes with a score above a certain threshold (or the top $M \ll N$ boxes with the highest score) are output by the detector.

3.2 Output Calibration using Relaxed Softmax

Even though the softmax operator $\sigma$ by definition ensures that the output score $s_i$ are in the interval $(0, 1)$, it has been empirically shown [24, 8] that it does not result in a good estimator of the posterior detection probability $p(y_i|I)$. In other words, the softmax output does not reflect well the model confidence; in practice, deep networks tend to be overconfident in their predictions (see Figure 2a), i.e. the scores associated with the detections tend to be higher than the true probability of a correct detection.

Next, we propose a method that encourages the model output $s_i$ to be as close as possible to the posterior $p(y_i|I)$. Based on the observation that networks tend to output high scores even for hard examples, where in principle the confidence should be lower, we let our model express its uncertainty by introducing an additional parameter to the softmax operator that lets the network “soften” its predictions for particular samples. We build on the model of temperature scaling [11, 9] for its simplicity, where the classification output is given as

$$\sigma_i(j; T) = \frac{\exp(\frac{z_{i,j}}{T})}{\sum_{k=1}^{K+1} \exp(\frac{z_{i,k}}{T})}.$$ \hspace{1cm} (4)

For $T > 1$, this allows the model to “soften” its predictions and with $T \to \infty$ the score for all classes is identical. For $T = 1$, we obtain the original softmax operator (1).

Next, we modify eq. (2) to obtain our relaxed softmax layer. The first consideration is to allow the network to learn to self-adjust the temperature on a sample-by-sample basis, therefore replacing a global parameter $T$ with a sample-dependent temperature $T_i$. Equation (4) however is not well-suited for loss optimization
through SGD due to numerical instabilities when \( T_i \to 0 \). We therefore propose an equivalent formulation, which we refer to as \textbf{relaxed softmax}, which is more numerically stable and can therefore easily be plugged in to standard loss optimization frameworks:

\[
\hat{s}_i(j) = \frac{\exp(\alpha_i z_{i,j})}{\sum_{k=1}^K \exp(\alpha_i z_{i,k})}, \quad \alpha_i := \frac{1}{T_i},
\]

(5)

In this model, the network \( \Phi(R_i; I) \) predicts a vector \((z_{i,1}, \ldots, z_{i,K}, \alpha_i)\) for each candidate box \( R_i \), and the candidate box is then associated with a score \( \hat{s}_i \) using eq. (5). Note that from computational perspective, this is equivalent to predicting \( K + 1 \) scalar scores instead of \( K \), which is a negligible cost.

We think, that because in our model the network can in fact improve its training objective by outputting “not so certain” decisions \((T > 1)\) for ambiguous samples, it is not forced to be over-confident like the standard models where even the ambiguous samples are required by the loss function to output 1.0 confidence. And on contrary, for non-ambiguous samples, the training objective is still maximized by setting \( T = 1 \), because this puts the probability mass entirely behind the correct class. These two opposing effects, we believe, help the model to become more calibrated, because it allows them to learn and infer how ambiguous the (training) samples are.

### 3.3 Evaluation Metrics

In order to visually quantify the quality of the calibration of a probabilistic classifier, we adopt the \textbf{Reliability Diagrams} of \cite{guo2017calibration}. A reliability histogram shows the score \( s_i \) generated by a method along the abscissa, plotting it against the actual probability of correct detection \( p(y_i|I) \). Since the true posterior \( p(y_i|I) \) is unknown, we empirically approximate the posterior \( \hat{p}(y_i|I) \) as a fraction of correctly classified test samples, and compare the network’s output \( s_i \) to the latter. For example, given 100 samples, each with a score \( s_i = 0.6 \), we would expect 60 samples to be correctly classified and 40 samples to be classified incorrectly.

In the diagram, if the bin which corresponds to the output score 0.6 contains 40% correctly classified samples then the model is \textit{overconfident}, and in contrast, if it contains 80% the model is \textit{conservative}. The desired outcome therefore is that in each bin the method’s output score corresponds to the true ratio of correct samples, which is when refer to the method as \textbf{calibrated}.

Formally, given the method’s output as a set of pairs \((s_i, y_i)\) of predicted scores \( s_i \) and labels \( y_i \), the corresponding set of ground truth labels \( \hat{y}_i \), and the number of bins \( M \), the \textbf{accuracy} \( A_m \) is defined as

\[
A_m = \frac{1}{|B_m|} \sum_{i \in B_m} \mathcal{I}(y_i = \hat{y}_i), \quad \text{where} \quad B_m = \left\{ i : \frac{m - 1}{M} < s_i \leq \frac{m}{M} \right\}
\]

(6)

where \( \mathcal{I} \) is the indicator function. We also define an \textbf{average score} \( S_m \) for the bin \( m \) as

\[
S_m = \frac{1}{|B_m|} \sum_{i \in B_m} s_i
\]

(7)

A reliability diagram has \( S_m \) on the abscissa and \( A_m \) on the ordinate.

Given the reliability diagram \((S_m, A_m)\), we introduce the \textbf{Average Calibration Error} (ACE) measure as the primary metric for object detection calibration, as the existing Expected Calibration Error (ECE) metric is not suitable for object detectors (see below). ACE measures the average absolute difference between the score and the accuracy in all bins as

\[
ACE = \frac{1}{M^+} \sum_m |S_m - A_m|
\]

(8)

where \( M^+ \) is the number of non-empty bins. ACE therefore assigns an equal weight to each bin, which is desirable for safety-critical applications where we want to measure the total deviation between the predicted score and the accuracy, irrespective to how frequently the object can appear in the image.

We also measure output calibration using the \textbf{Expected Calibration Error} (ECE) \cite{guo2017calibration}, which weights the error based on the number of samples in each bin:

\[
ECE = \sum_m \frac{|B_m|}{n} |S_m - A_m|
\]

(9)

This metric however is not very well-suited for the evaluation of object detectors, because there are typically many low-confidence detections with a score close to 0, which gives disproportionate weight (typically more than 95%) to the first bin, so the resulting ECE value is then mostly given by the error in the low-confidence predictions.

Finally, we report the \textbf{Maximum Calibration Error} (ECE) \cite{guo2017calibration}, as this is also very relevant for robust object detection, because it empirically measures the maximal error a method can make in the probability estimation. ECE is given by:

\[
MCE = \max_m |S_m - A_m|
\]

(10)
In order to compare the traditional softmax to the introduced relaxed softmax formulation, we evaluate their properties on two large-scale public benchmarks, using the following experimental setup: We use the state-of-the-art pedestrian detector SDS-RCNN [2] and train two models from scratch — the first model uses softmax (same as [2]), whereas the second model uses relaxed softmax for pedestrian/background classification in both stages of the SDS-RCNN. For each model, we also experiment with a subsequent parametric calibration of its output $s_i$, using the linear $s_i'$ and temperature scaling $s_i''$ of Guo et al. [9], given by:

$$s_i' = \min \{1, \beta s_i\}$$

$$s_i'' = \frac{\exp(\frac{s_i}{T})}{\sum_{k=1}^{K} \exp(\frac{z_k}{T})}$$

where $\beta$ (respectively $T$) is a single global hyper-parameter, whose optimal value for the given trained model was found on the validation subset by minimizing the Average Calibration Error (ACE). When it is used, linear/temperature scaling is tuned on the validation data after learning the model on the training data.

Next, we show results on the Caltech and NightOwls datasets, comparing the calibration of the output scores $s_i$, $s_i'$ and $s_i''$ on the respective testing set using the metrics of Section 3.3.

### 4.1 Caltech Dataset

The Caltech dataset [5] is the most-commonly used dataset for pedestrian detection. For training, we use the same setting as in [2]: the learning rate is set to $10^{-3}$ and then it is dropped by a factor 10 after every 60,000 iterations. We train the network by minimizing eq. 5 wrt. to $\alpha_i$ and $z_{i,j}$ using vanilla SGD for 180,000 iterations.

We observe that the network which uses softmax is over-confident (see Figure 2a), as the output score is higher than the real ratio of true positives. We also see that linear scaling (11) has little effect on output calibration (see Figure 2b) and that the temperature scaling (12) causes the network to be calibrated only in the center of the (0, 1) confidence interval, but to underestimate number of true positives for scores above 0.5 and inversely to over-estimate the number of true positives below 0.5 (see Figure 2c). This is because temperature scaling only relies on a single global parameter, and therefore the only way to improve the calibration is globally increasing the entropy of the scores distribution [9].

Using the relaxed softmax formulation, on the other hand, causes the network to be monotonically more conservative by under-estimating the number true positives in each bin (see Figure 2d) — we think this under-estimation is caused by the fact there are not that many ambiguous training samples in the dataset. Thanks to the monotonicity of the difference, simple linear scaling (11) however makes sure the network becomes almost perfectly calibrated (see Figure 2e). As a result, the method output score can be interpreted as the model confidence.

Quantitatively, the relaxed softmax has lower Average Calibration Error (ACE), Expected Calibration Error (ECE) and Maximum Calibration Error (MCE) than softmax and softmax with linear scaling, but has higher ACE than softmax with temperature scaling [9]. Relaxed softmax with linear scaling, however, outperforms all other formulations by a significant margin, lowering the ACE 4 times and MCE 3 times when compared to the commonly used softmax (see Table 1).

The average miss rate (using the “reasonable” setting of [6]) of relaxed softmax is however slightly worse than standard softmax — we speculate that this is because in training the network “gives up” on certain hard samples, or gives them a lower confidence score, which also negatively impacts the average miss rate, as...
Table 1: Average Calibration Error (ACE), Expected Calibration Error (ECE), Maximal Calibration Error (MCE) and average Miss Rate of the SDS-RCNN pedestrian detector on the Caltech dataset, using different classification output formulations.

<table>
<thead>
<tr>
<th>Classification Output</th>
<th>ACE</th>
<th>ECE</th>
<th>MCE</th>
<th>Miss Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax</td>
<td>20.62 %</td>
<td>1.86 %</td>
<td>38.55 %</td>
<td>10.17 %</td>
</tr>
<tr>
<td>Softmax + linear scale</td>
<td>18.52 %</td>
<td>1.81 %</td>
<td>35.26 %</td>
<td>10.17 %</td>
</tr>
<tr>
<td>Softmax + temperature scale</td>
<td>9.58 %</td>
<td>1.95 %</td>
<td>18.49 %</td>
<td>10.17 %</td>
</tr>
<tr>
<td>Relaxed Softmax</td>
<td>14.72 %</td>
<td>0.38 %</td>
<td>29.95 %</td>
<td>13.26 %</td>
</tr>
<tr>
<td>Relaxed Softmax + linear scale</td>
<td><strong>5.62 %</strong></td>
<td><strong>0.13 %</strong></td>
<td><strong>14.73 %</strong></td>
<td><strong>13.26 %</strong></td>
</tr>
</tbody>
</table>

Table 2: Average Calibration Error (ACE), Expected Calibration Error (ECE), Maximal Calibration Error (MCE) and average Miss Rate of the SDS-RCNN pedestrian detector on the NightOwls dataset.

<table>
<thead>
<tr>
<th>Classification Output</th>
<th>ACE</th>
<th>ECE</th>
<th>MCE</th>
<th>Miss Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax</td>
<td>17.67 %</td>
<td>4.34 %</td>
<td>33.99 %</td>
<td>19.85 %</td>
</tr>
<tr>
<td>Softmax + linear scale</td>
<td>15.37 %</td>
<td>3.75 %</td>
<td>30.08 %</td>
<td>19.85 %</td>
</tr>
<tr>
<td>Softmax + temperature scale</td>
<td>12.09 %</td>
<td>0.12 %</td>
<td>24.28 %</td>
<td>19.85 %</td>
</tr>
<tr>
<td>Relaxed Softmax</td>
<td><strong>7.56 %</strong></td>
<td><strong>0.07 %</strong></td>
<td><strong>18.84 %</strong></td>
<td><strong>20.62 %</strong></td>
</tr>
<tr>
<td>Relaxed Softmax + linear scale</td>
<td>7.74 %</td>
<td><strong>0.07 %</strong></td>
<td>24.58 %</td>
<td><strong>20.62 %</strong></td>
</tr>
</tbody>
</table>

more false positives will be ranked higher than such samples. Note that the miss rate is not affected by the subsequent parametric calibration, as both transformations are a monotonically increasing function. We can say that the network is slightly worse at guessing, but it is more reliable, which is likely a good trade off in applications.

4.2 NightOwls Dataset

The NightOwls dataset contains 279,000 night images recorded in 3 countries by an industry-standard camera mounted onto a moving car. The dataset is captured at night, at dusk or at dawn, it contains different seasons and weather conditions and is therefore significantly more challenging than the Caltech dataset. It also by definition contains more ambiguous scenarios than the Caltech dataset (see Figure 1), because of low contrast and severe motion blur, caused by high exposure times of the camera. We train again train the SDS-RCNN detector on the training subset for 300,000 iterations using the learning rate of $10^{-3}$, which is dropped by the factor of 10 after every 100,000 iterations, and we evaluate the output calibration on the testing subset.

We observe that softmax is again over-confident in its predictions, and unlike for the Caltech dataset, the subsequent calibration on the validation set using linear or temperature scaling does not seem to help (see Figure 2a-c). In contrast, the proposed relaxed softmax is significantly better calibrated, even without any subsequent output scaling on the validation set (see Figure 2d and Table 2). We suggest this is because there are more ambiguous training samples in the NightOwls than in the Caltech dataset, which allows the network to learn from samples distributed over the whole $(0, 1)$ confidence interval.

The linear scaling of the output scores then actually makes the calibration error marginally worse (see Figure 2e), which we speculate is due to subtle differences in the data statistics between the validation and the testing set.

5 Conclusion

Having a confidence measure that accurately reflects a neural network’s prediction reliability is a crucial requirement for many applications, especially in safety-critical scenarios such as autonomous driving. In this paper, we presented a framework for the systematic assessment of detectors in terms of the quality of their confidence estimates and we showed that current state-of-the-art pedestrian detection methods are over-confident in their predictions when the evidence in the data is genuinely scarce or ambiguous.

Additionally, we proposed a simple but effective modification of the standard softmax layer called relaxed softmax, that significantly improves the detectors self-calibration properties. To achieve this, the network has to estimate just one more scalar output per sample, which is of negligible computational impact in all practical architectures. We evaluated its (Average) Calibration Error on two large-scale pedestrian detection datasets, and demonstrated that relaxed softmax leads to better output calibration than the pre-existing methods.

We note that relaxed softmax is a drop-in replacement that can be used in any scenario where softmax and cross entropy are used in learning deep neural networks. While we did not yet experiment with other scenarios, the benefits we have observed are likely to transfer to many other cases as well.
References


