Reliability Scoring for the Recognition of Degraded License Plates*

Anatol Maier
Friedrich-Alexander University
Erlangen-Nürnberg, Germany
anatol.maier@fau.de

Denise Moussa
Federal Criminal Police Office (BKA)
Wiesbaden, Germany
denise.moussa@fau.de

Andreas Spruck
Friedrich-Alexander University
Erlangen-Nürnberg, Germany
andreas.spruck@fau.de

Jürgen Seiler
Friedrich-Alexander University
Erlangen-Nürnberg, Germany
juergen.seiler@fau.de

Christian Riess
Friedrich-Alexander University
Erlangen-Nürnberg, Germany
christian.riess@fau.de

Abstract

Criminal investigations oftentimes need the identification of license plates of escape vehicles. The vehicles may be recorded by low-quality cameras in the wild. Their license plates may be unreadable for police officers. Recent efforts aim to use machine learning to forensically decipher license plates from such low-quality images. These methods operate near the information-theoretic limit of recognition and hence show quite high error rates. Unfortunately, it is unclear when such prediction errors occur, which makes it difficult to use these methods in practice. In this work, we propose a Bayesian Neural Network to inherently incorporate a reliability measure into the classifier. We additionally propose to integrate multiple estimations with an entropy weight to further improve the reliability. Our experiments show that this uncertainty metric dramatically reduces the number of false predictions while preserving most of the true predictions.

1. Introduction

License plates recordings on photo and video are often an important cue in a criminal investigation. They may serve as forensic trace or be probative in a legal setting. However, these recordings oftentimes come from uncontrolled devices and uncontrolled acquisition conditions. Hence, they are oftentimes of low quality, e.g., due to strong compression, bad camera optics, large vehicle distance, and environmental factors like rain. When the license plates are unreadable for humans, the goal of forensic license plate recognition (FLPR) is to reconstruct the characters of the license plate. Even only partly reconstructed license plates can already help to identify and hold potential perpetrators accountable for their actions [3]. Though the reconstruction only serves as an investigative clue for an analyst, not as evidence in court.

From a technical perspective, FLPR systems have to fulfill two key requirements. First, they have to generalize across a wide range of scenes, acquisition devices, illumination conditions, and noise artifacts. Second, a robust and reliable prediction is an integral element. In recent years, deep neural networks gained high popularity also for license plate detection and classification [24, 23, 1, 14]. These results have shown that specially trained convolutional neural networks (CNNs) are able to outperform human performance for character recognition of highly degraded license plates. However, these methods do not give much insights about their decision making, which makes it difficult to understand when the models fail. For example, if exposed to so-called out-of-distribution examples, these models might fail while making confident decisions [16]. This poses a severe challenge in a forensic context, where reliability and a graceful decline in performance are important require-

*We gratefully acknowledge support of our work by the German Federal Ministry of Education and Research (BMBF) under grant number 13N15319.

978-1-6654-6382-9/22/$31.00 ©2022 IEEE

ments.

A way to improve the reliability of models is to explicitly include a notion of uncertainty into the system. An uncertainty measure enables an operator to decide whether she can trust the result or not. Neural networks are also capable of providing decisions with an additional uncertainty measure. To this end, the commonly used softmax outputs are replaced by probability distributions. Blundell *et al*. proposed such a network design by utilizing a variational approximation approach [2]. However, the resulting predictive uncertainty alone might not be a sufficient indicator for a reliable prediction. In cases of high entropy over the predictions, the expressed uncertainty is not meaningful. To counter this issue, we propose a reliability measure that incorporates the predictive uncertainty and the predictive entropy together with domain-specific prior information.

In this work, our contribution is threefold. First, we propose a Bayesian neural network (BNN) for for license plate recognition. Second, we propose to model the conditional dependence between consecutive license plate frames with an entropy weight. Third and most importantly, we propose a reliability score that integrates predictive uncertainty, predictive entropy, and domain-specific prior information. We show in our experiments that the proposed entropy weight improves the reliability for both models compared models. The proposed reliability score from our third contribution works best in combination with our BNN. The reliability score enables the rejection of most false decisions while preserving most of the true decisions.

The remainder of this paper is organized as follows. Section 2 describes related work on uncertainty modeling and on recognition of degraded license plates. Section 3 introduces the Bayesian neural networks, the proposed entropy weighting, and the proposed reliability score. Section 4 presents our experiments, and Sec. 5 concludes the work.

2. Related Work

The sensitivity of machine learning models to out-of-distribution samples is of increasing importance in reliability-critical application fields like autonomous driving [17] or multimedia forensics [15, 16]. Hendrycks and Gimpel propose to detect out-of-distribution examples , and hence to anticipate misclassifications by analyzing the classifier softmax activations [8]. However, Guo *et al*. showed that these softmax statistics do not represent the likelihood of correctness well [7]. Softmax statistics can be improved via calibration to the model confidence. For example, Liang *et al*. proposed the temperature scaling method [13]. However, this approach may still lead neural networks to make overly confident decisions in out-of-distribution domains as shown by Maier *et al*. [16].

Various approaches address these issues by enabling a neural network to express uncertainty in its predictions. One approach is to explicitly learn confidence estimates in a neural network with two output branches that provide the prediction and its uncertainty estimate [4]. Another approach is to approximate the posterior distribution in a Bayesian framework. Gal and Ghahramani proposed MC-dropout as a discrete approximation [6]. Lakshminarayanan *et al*. propose a similar idea, namely to use an ensemble of neural networks for uncertainty estimation [11]. Both approaches, however, only approximate the full posterior distribution of the Bayesian approach. Blundell *et al*. [2] show that variational approximation can be utilized for neural networks. The resulting Bayesian neural network (BNN) models probability distributions instead of scalar point estimates over the trainable parameter space. This property enables the network to predict uncertainty, which allows to assess the reliability of a prediction. BNNs have shown solid performance, robustness, and reliability in various tasks, *e.g*., pixel-wise depth regression [9], biomedical image segmentation [10] and multimedia forensics [15, 16]. In this work, we apply BNNs to the reliability-critical field of forensic license plate recognition.

Early works on license plate recognition oftentimes assume controlled acquisition conditions, and hence data of relatively high quality. Early works propose handcrafted character recognition pipelines [19, 20, 26]. Li and Shen [12] introduced the first CNN architecture with bi-directional recurrent neural network and a connectionist temporal classification (CTC). A similar approach by Zou *et al*. proposes a Bi-LSTM [28].

There is an increasing number of works that address license plate recognition from low-quality images. Špaňhel *et al*. propose a CNN for the recognition of low resolution and low quality European license plates [24]. Agarwal *et al*. was first to propose a CNN for deciphering unreadable US license plates [1]. Lorch *et al*. extended this work [14], and Moussa *et al*. proposed a sequence-based method for the same task [18]. Additionally, Rossi *et al*. [22] extended the work of Lorch *et al*. [14] by combining a UNet-based [21] denoising network with a license plate deciphering CNN. Since the proposed two staged model by Rossi *et al*. [22] performs on par with Lorch *et al*. [14], we use the CNN architecture by Lorch *et al*. as our baseline model to recognize low-quality German license plates. In similar spirit to the work of Rossi *et al*., our proposed method supports an analyst with additional information. However, in contrast to previous works, our main focus is on reliable estimates. Our proposed method integrates multiple individual estimates, that can either come from a sequence of images or from a set of BNN estimates from a single image.

3. Reliable License Plate Recognition

Training a standard neural network consists of learning optimal parameters, or point estimates, that maximize the

unknown posterior distribution $P(\boldsymbol{\omega}|\mathcal{D})$ over the weights $\boldsymbol{\omega}$ given some training data $\mathcal{D}$. In contrast to this paradigm, Bayesian deep learning aims to estimate the full posterior distribution over the weights. This enables the estimation of the predictive uncertainty on unseen data. Exact inference is intractable due the large parameter space within a typical neural network (which is not surprising considering that exact inference is also intractable in much smaller Bayesian models). However, the posterior distribution can be approximated via variational inference, as shown by Blundell *et al*. [2]. Through variational approximation, the intractable integration problem is reformulated into an optimization problem. Here, the goal is to find optimal parameters θ of the weight distribution $q(\boldsymbol{\omega}|\theta)$, subject to minimizing the Kullback-Leibler divergence between $q(\boldsymbol{\omega}|\theta)$ and the true unknown distribution $P(\boldsymbol{\omega}|\mathcal{D})$.

$$\begin{aligned}\theta^* &= \underset{\theta}{\operatorname{argmin}}\, \mathrm{KL}\left[q(\boldsymbol{\omega}|\theta)||P(\boldsymbol{\omega}|\mathcal{D})\right] \\ &= \underset{\theta}{\operatorname{argmin}}\, \mathrm{KL}\left[q(\boldsymbol{\omega}|\theta)||P(\boldsymbol{\omega})\right] - \mathbb{E}_{q(\boldsymbol{\omega}|\theta)}\left[\log P(\mathcal{D}|\boldsymbol{\omega})\right] \quad .\end{aligned} \tag{1}$$

Once we obtain the variational parameters, we can approximate the exact cost as described by Blundell *et al*. [2] and formulate an estimator of the predictive posterior through sampling from the variational posterior distribution, according to Kwon *et al*. [10] as

$$\begin{aligned}\mathbb{E}_{q(\boldsymbol{\omega}|\theta)}\left[P(\boldsymbol{y}^*|\boldsymbol{x}^*)\right] &= \int P(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\omega}) q(\boldsymbol{\omega}|\theta) d\boldsymbol{\omega} \\ &\approx \frac{1}{n}\sum_{i=1}^{n} P_{\boldsymbol{\omega}^i}(\boldsymbol{y}^*|\boldsymbol{x}^*) \quad ,\end{aligned} \tag{2}$$

with $\boldsymbol{x}^*$ representing an unknown sample, $\boldsymbol{y}^*$ being the predicted class and $P_{\boldsymbol{\omega}^i}(\boldsymbol{y}^*|\boldsymbol{x}^*)$ denoting sampling from the predictive posterior. The approximation in Eq. 2 draws n Monte-Carlo samples from the trained network on unseen data. Then the network's uncertainty over the predictions is expressed via the variance of our estimator [10],

$$\mathrm{Var}\left[P(\boldsymbol{y}|\boldsymbol{x})\right] = \mathbb{E}_{q(\boldsymbol{\omega}|\theta)}[\boldsymbol{y}\boldsymbol{y}^T] - \mathbb{E}_{q(\boldsymbol{\omega}|\theta)}[\boldsymbol{y}]\mathbb{E}_{q(\boldsymbol{\omega}|\theta)}[\boldsymbol{y}]^T \quad . \tag{3}$$

By evaluating the predictive uncertainty for an unseen data sample, we can consider the uncertainty as a proxy for reliability. The predictive uncertainty is a measure of the disagreement between different instances of the BNN. A higher uncertainty implies various possibilities to explain the given data sample, due to the BNN's inability to extrapolate to the data distribution from which the sample is drawn. However, especially in high entropy regions, the uncertainty alone is a poor indicator of reliability. If the model output is an uninformative prediction, i.e., when each class is equally likely, the predictive uncertainty is extremely low and thus also uninformative. We can account for this caveat

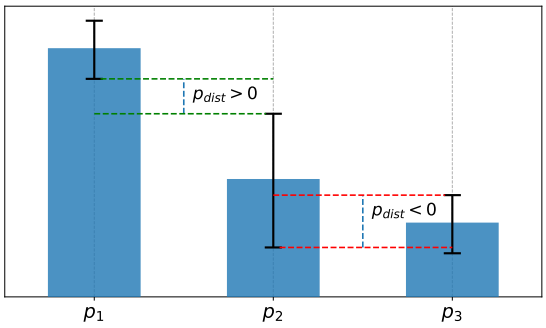


Figure 1: Schematic visualization of the calculation of the predictive distance between two predictions. The blue bars p_1, p_2, p_3 representing example mean predictions, whereas the error bars represent the network's uncertainty over the predictions. A higher distance is reflected in an increased reliability score, while a smaller or negative distance will reduce the score, respectively.

by also considering the entropy over the predictions. The entropy is defined as

$$H(\boldsymbol{p}) = -\sum_{i=1}^{C} p_i \cdot \log(p_i) \tag{4}$$

where C denotes the number of classes. Hence, the most uninformative prediction is $p_1 = p_2 = \cdots = p_C$, thus the maximum entropy is then given as

$$\begin{aligned}H_{\max}(\boldsymbol{p}) &= -\sum_{i=1}^{C} \frac{1}{C} \cdot \log(\frac{1}{C}) \\ &= \log(C) - \log(1).\end{aligned} \tag{5}$$

Based on Eq. 4 and Eq. 5, we define a weighting term that expresses the predictive entropy as

$$I(\boldsymbol{p}) = 1 - \frac{H(\boldsymbol{p})}{H_{\max}(\boldsymbol{p})}. \tag{6}$$

Equation 6 will also be used as entropy weight in the evaluation. Further, instead of evaluating the overall uncertainty over the predictions, we evaluate the predictive distance $D_p(i,j)$ between two predictions, which we define as

$$\begin{aligned}D_p(i,j) =& \mathbb{E}_{q(\boldsymbol{\omega}|\theta)}\left[P(y_i^* = 1|\boldsymbol{x}^*)\right] - \mathrm{Var}\left[P(y_i^* = 1|\boldsymbol{x}^*)\right] - \\ & \mathbb{E}_{q(\boldsymbol{\omega}|\theta)}\left[P(y_j^* = 1|\boldsymbol{x}^*)\right] - \mathrm{Var}\left[P(y_j^* = 1|\boldsymbol{x}^*)\right]\end{aligned} \tag{7}$$

and normalize into $[0, 1]$ range using the sigmoid function

$$\sigma(D_p(i,j)) = \frac{1}{1 + \mathrm{e}^{(-\alpha \cdot D_p(i,j))}}, \tag{8}$$

where we empirically set $\alpha = 6$. A visual explanation of the predictive distance is shown in Fig. 1. Here the blue bars p_1, p_2, p_3 represent three example mean predictions of one of the output layers. The error bars represent the network's uncertainty over the predictions, thus the variance as defined in Eq. 3. The predictive distance is then defined as the overlap between the mean prediction and the respective variance. A higher absolute distance with $p_{\text{dist}} > 0$ is reflected in an increased reliability score. Whereas a smaller or negative distance, represents the networks inability to distinguish which prediction is more likely. In the example case this would be the distinction between p_2 and p_3, thus reflected in a decreased reliability score.

The reliability score is defined as

$$R(\boldsymbol{p}, D_p(i,j)) = \exp\left(\beta \cdot \frac{I(\boldsymbol{p})}{\sigma(D_p(i,j))}\right), \qquad (9)$$

where we empirically set $\beta = -10$.

4. Experimental Results

We conduct a series of experiments on license plate recognition on real world video data to evaluate the Bayesian CNNs (BNN) ability to express uncertainty and its respective reliability. To this end we compare our proposed BNN model to the state-of-the-art model for severely degraded license plate recognition by Lorch *et al.* [14]. The CNN by Lorch *et al.* [14] is originally trained on Czech license plates. However, we slightly adapt the model to our task and retrain it on a synthetic dataset of German license plates for a direct and fair performance comparison.

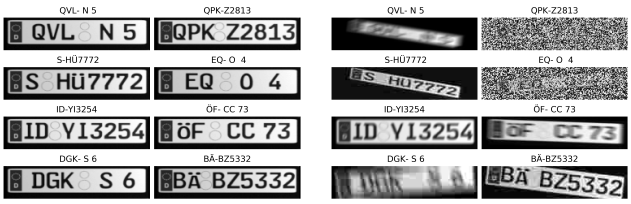
4.1. Datasets

The training is performed on a synthetic dataset of German license plates. Here, we can apply strong degradations (e.g., down-sampling, blur, and compression) in a controlled way via data augmentation. For the evaluation, we record a smaller set of real images of German license plates on moving vehicles. Both datasets consist of a sequence of characters followed by numbers. German area codes are replaced by random characters to prevent biases towards license plates from large cities.

4.1.1 Synthetic Dataset

The synthetic dataset consists of about 1.3 million images of German license plates. The images are created with size 180×40 pixels with the official font and formatting using the 3D rendering framework by Spruck *et al.* [25]. Examples are shown in Fig. 2 (a).

Extensive augmentations are applied during training with Gaussian noise, motion blur, random rotation, random padding, down-sampling, and JPEG-compression. The parameter ranges per augmentation type are listed in Tab. 1. Most augmentations are applied independently with probability $p_{\text{aug}} = 0.5$. One exception is motion blur, where we empirically chose $p_{\text{aug}} = 0.7$ after noticing that this augmentation has a stronger impact on the performance. The other exception is that if Gaussian noise or JPEG compression are selected, the image is also downsampled prior to the noise or JPEG augmentations. Finally, each image is again up-sampled to the original resolution of 180×40. For the down- and up-sampling operation, we randomly decide between bilinear-, bicubic-, nearest neighbor-, areal- and Lanczos interpolation method. Figure 2 (b) shows example augmentations. The synthetic dataset is split into distinct sets of 80% for training, 10% for validation, and 10% for testing.



(a) Generated synthetic data (b) Augmented samples

Figure 2: (a) Example images from the synthetic German license plate dataset. (b) License plate images after data augmentations for the model training.

Table 1: Augmentation parameters for training (see text for details).

Augmentations	**Prob.**	**Min.**	**Max.**
Gaussian Noise	0.5	$\mu = 0$ $\sigma = 0.001$	$\mu = 0$ $\sigma = 0.5$
Motion Blur	0.7	$\alpha = 1°$ kernel = 2	$\alpha = 180°$ kernel = 30
Rand. Rotation	0.5	$\alpha = 1°$	$\alpha = 10°$
Rand. Padding	0.5	1 px	50 px
JPEG Comp.	0.5	$q = 5$	$q = 99$
Downsampling	0.5	width = 20	width = 160

4.1.2 Real-world Dataset

The real-world dataset consists of 95 sequences of vehicles that pass by a Google Pixel 2 smartphone camera. All scenes show a total of 15 637 frames with 164 frames on average per scene. Each license plate is rendered, using the same pipeline as for the synthetic dataset, printed on paper and attached manually to the car on top of the real license plate. The recordings show in each scene an approaching

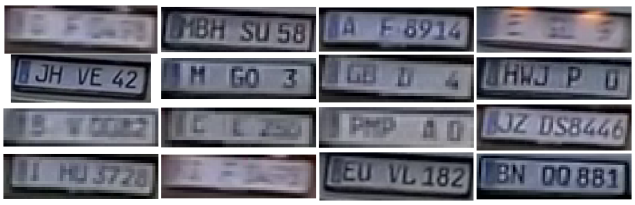


Figure 3: Example license plate images from the real-world dataset.

or passing car. Figure 3 shows example images from the real-world dataset.

4.2. Model Architecture

All models are implemented in Tensorflow. The CNN by Lorch *et al*. [14] serves as backbone. The model expects a grayscale image with resolution of 180×40 followed by four convolution blocks. Each convolution block consists of two convolution layers with a receptive field of 3×3, followed by a maximum pooling operation. The classification head consists of two fully connected layers with 2048 and 512 units, respectively, followed by the output layers. We modify it to use nine output layers, each of which is a fully-connected layer with 41 nodes and softmax activation. The 41 nodes encode the characters a to z, 0 to 9, three German umlauts, space, and '-' to separate the area code from the other characters. Figure 4 shows a schematic overview of the architecture. The BNN uses the CNN as a template, i.e., with identical numbers of layers, filter kernel dimensions per convolution layer, and fully-connected layers. To make it Bayesian, we replace the fully-connected layers by flipout fully-connected layers [27] from the Tensorflow probability framework [5]. The prior distribution is heuristically chosen as a zero-mean Gaussian with unit-variance. We perform inference via Eq. 2 with $n = 100$ Monte Carlo samples, and obtain the predictive variance from Eq. 3. We assume a normally-distributed variational posterior, thus the BNN has roughly twice as many training parameters as the CNN.

4.3. Training Protocol and Evaluation Metrics

The CNN and the BNN are trained on the synthetic data. Both models are trained on augmented grayscale input images with a resolution 180×40, depicting the full license plate, as shown in Fig.2 (b). Each of the nine output layers is then trained to classify the character at the respective position within an image. The optimizer for training both models is Adam with a learning rate of $l = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-7}$, and a batch size of 128. We train the CNN for 150 epochs and our BNN for 250 epochs due to the larger number of parameters. We select the model by Lorch *et al*. [14] and the proposed BNN model from the epoch in which each performs best on the validation set.

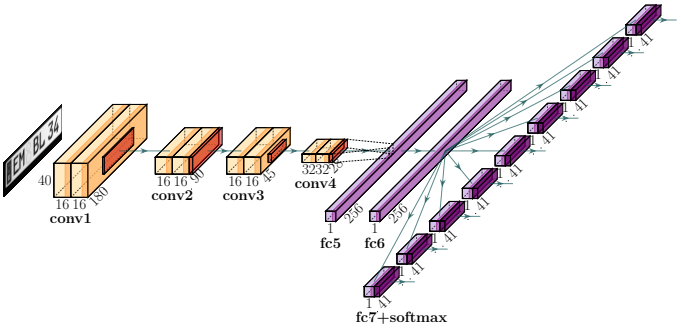


Figure 4: Model architecture for German license plate recognition. As input the model expects a fixed-size grayscale input image with resolution 180×40, followed by four convolution blocks. The classification head consists of two fully connected layers and nine output layers each with 41 output nodes followed by a softmax activation.

These models are used for the evaluation.

We report two metrics in the evaluation. First, the per-character accuracy, i.e., the ratio of correctly classified characters in the test set. Second, the license plate accuracy, i.e., the ratio of the correct classification of a whole license plate over all license plate images. Additionally, we report the top-k accuracies for $k \in \{1, 3, 5\}$ where we count whether the true answer is within the k most likely predictions.

4.4. Recognition Accuracy on Synthetic Data

Both models perform strongly when testing on synthetic images, i.e., when training and test distributions are well aligned. The top-1 per-character accuracies are 0.920 and 0.923 for the CNN and our BNN, respectively. Per-license plate accuracies are 0.747% and 0.755% for the CNN and the proposed BNN. Hence, both models perform approximately equally well when the training and test data distribution are aligned. However, we note that this assumption is (almost certainly) overly optimistic, since images from real cases come from entirely uncontrolled sources.

4.5. Recognition Accuracy on Out-of-Distribution Real Data

We now focus on the test set of real images, which is entirely unseen, i.e., no data from this source has been used during training. This introduces a significant shift between the training and testing distributions, which can also visually be observed when comparing Fig. 2 with Fig. 3. The difference between training and testing distribution considerably reduces the absolute accuracies. Nevertheless, we argue that police investigations are oftentimes forced to operate under such a distribution shift, since the data under investigation almost always comes from third parties, and

Table 2: Accuracy per character on real data. The best performing according to the applied weighting is marked bold for each model. The entropy weighting greatly improves the model performance for both the CNN by Lorch and the BNN.

Weighting	Net	Top-1	Top-3	Top-5
independent	Lorch	0.369	0.531	0.628
	BNN	0.374	0.531	0.626
uniform	Lorch	0.523	0.751	0.830
	BNN	0.542	0.757	0.828
entropy (proposed)	Lorch	**0.620**	**0.782**	**0.847**
	BNN	**0.627**	**0.794**	**0.850**

Table 3: Accuracy per license plate on real data. The best performing according to the applied weighting is marked bold for each model. Here, the entropy weight also shows significant improvements in license plate recognition rate for both models.

Weighting	Net	Top-1	Top-3	Top-5
independent	Lorch	0.007	0.054	0.095
	BNN	0.014	0.047	0.088
uniform	Lorch	0.011	0.189	0.305
	BNN	0.021	0.137	0.316
entropy (proposed)	Lorch	**0.032**	**0.284**	**0.389**
	BNN	**0.053**	**0.263**	**0.368**

we argue that it is infeasible to cover the space of all possible source distributions in the training process.

We evaluate three different weighting approaches to fuse additional information from several frames of video sequences. These weights are evaluated for the CNN by Lorch [14] and the proposed BNN. The weighting "independent" treats all inputs independently, i.e., without performing any inference across images. The weighting "uniform" integrates multiple estimates on the same license plate via averaging, i.e., as uniformly weighted estimates. The weighting "entropy" combines multiple estimates with the proposed entropy weighting.

Table 2 shows the per-character accuracies of these variants. We report a continuous increase in top-k accuracies between the three variants. The proposed entropy weighting achieves the highest accuracies for both evaluated networks. The relative performance between the CNN by Lorch and the proposed BNN is approximately on par. The relative improvement between the three weighting variants is substantial. For example, the top-1 accuracy increases from 0.369 to 0.523 when using uniform weighting, and further to 0.620 with the proposed entropy weight.

Analogous observations can be made for per-license plates accuracies in Tab. 3. Also here, the proposed entropy weighting outperforms the other weightings by a large margin. When comparing the CNN by Lorch with the BNN, the top-1 accuracies are somewhat better for the BNN, top-3 and top-5 accuracies are somewhat better for CNN.

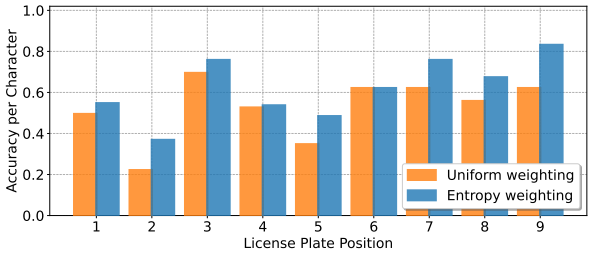


Figure 5: Recognition accuracy of uniform weights versus entropy weights per character position on the license plate. Entropy weights almost everywhere improve the accuracy.

The improvement of entropy weights over uniform weights is quite consistent, also across the individual positions of the license plate. Figure 5 shows that performance difference for each of the nine individual character positions. At almost all positions, the proposed entropy weights improve the accuracy.

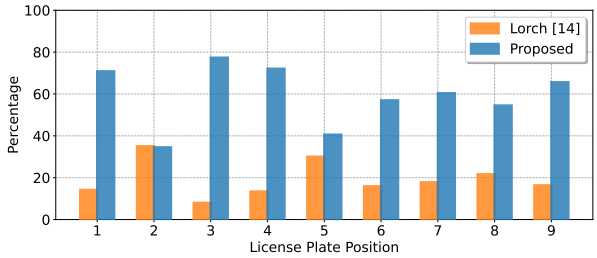
Overall, the entropy weight benefits from the fact that it prefers confident predictions with low entropy, and penalizes on uninformative predictions with high entropy. This approach leads to the substantial improvement in detection accuracy for all variants of top-k predictions.

4.6. Rejection of Unreliable Predicitons on Single Frames

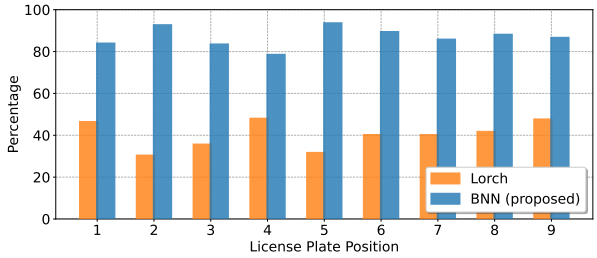
A practically relevant, but more challenging scenario is the case that only a single frame is available. This may be the case if a vehicle quickly passes by, and is recorded by a low-quality camera that stores the footage at a low framerate. The entropy weight cannot be used in this case, since it requires multiple frames that show the same license plate. Hence, the calculation of a reliability score can only use information from the inference within one input frame. The CNN only provides the softmax statistics to mimic a distribution of license plate characters. The BNN offers the additional advantage to inherently provide the predictive distribution from its Monte Carlo sampling, from which propose to calculate the entropy as reliability score (cf. Sec. 3).

We show its usefulness by rejecting uncertain decisions. More specifically, we show that thresholding on the reliability score enables the BNN to reject a large percentage of false predictions while preserving most of the correct predictions. For comparison, we perform the same experiment on the CNN's Softmax statistics.

The experimental protocol is described below. We first



(a) Preservation of correct predictions



(b) Reduction of false predictions

Figure 6: Performance comparison of Lorch [14] with Softmax statistics (orange) and our proposed BNN with reliability scoring (blue). The BNN preserves a higher percentage of correct classifications (top) and also reduces a higher percentage of false predictions (bottom). See text for details.

evaluate a baseline performance for the BNN and CNN, where we simply count the accuracy of predicting the most likely character per license plate position. Then, we threshold the reliability score to reject predictions with low reliability. In this experiment we set $r = 0.6$ as a tradeoff for rejecting false predictions and preserving correct predictions. Note that this threshold is not particularly optimized for the test data. An analyst can choose other tradeoffs (less false predictions or more correct predictions) by varying r.

Averaged over all real-world scenes and all character positions, the baseline CNN achieves an accuracy of 37.5% (i.e., with 62.5% false predictions), and the BNN achieves an accuracy of 37.9% (i.e., with 62.1% false predictions). Applying the threshold reduces the CNN's false predictions to 37.8% while reducing the correct predictions to 6.3%. Conversely, the proposed BNN achieves a reduction of its false predictions to 7.8% while preserving the correct predictions at 23.5%. We argue that the reduced accuracy is justified by the dramatic reduction of false predictions by almost 90%. These results are further illustrated per character position in Fig. 6. On top, we show the percentage of preserved correct classifications. On bottom, we show the percentage of reduced false predictions.

In summary, the CNN is not able to correctly identify wrong classifications as non-reliable predictions. Additionally, it does not preserve correct classifications. Conversely, the proposed BNN with reliability score shows significantly better performance on preserving correct classifications, while at the same time significantly reducing false classifications. On average, the CNN preserves 16% of correct classification while our method keeps 62%. Similarly, the CNN reduces the false predictions on average by only 39%, while our BNN reduces false predictions on average by 89%. Hence, the CNN rejects mostly correct classifications while false predictions are still mostly classified as reliable predictions.

Hence, the proposed reliability scoring further increases the reliability of the BNN classifier by a considerable margin. The false predictions are greatly reduced, while most of the correct classification results are preserved.

5. Conclusion

In this work we investigate approaches to enhance the reliability for recognizing severely degraded license plates. First, we use the conditional dependence among frames that show the same license plate. We propose an entropy weight to combine predictions from multiple frames by their confidence. This approach performs considerably better than a naive weighting or an independent treatment of the predictions. Second, we propose a Bayesian neural network that is able to express uncertainty even within only one single frame. Third, we propose a reliability metric that combines the entropy over the predictions with predictive uncertainty. It enables the network to detect and abstain from unreliable predictions. Removing unreliable predictions greatly increases the portion of correct prediction. Here, our proposed approach clearly outperforms Softmax statistics for reliable classification.

We hope that the proposed methods help towards closing an important gap for reliability-critical applications. In future work, we will expand the investigation of reliability enhancements to further model architectures and backbones like ResNet, DenseNet, or other CNN families as well as to alternative methods for expressing uncertainty.

References

[1] S. Agarwal, D. Tran, L. Torresani, and H. Farid. Deciphering severely degraded license plates. *Electronic Imaging*, 2017(7):138–143, 2017.

[2] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

[3] R. Böhme, F. C. Freiling, T. Gloe, and M. Kirchner. Multimedia forensics is not computer forensics. In *Computational Forensics*, pages 90–103. Springer Berlin Heidelberg, 2009.

[4] T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[5] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

[6] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[7] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[8] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations*, 2017.

[9] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

[10] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142, 2020.

[11] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[12] H. Li and C. Shen. Reading car license plates using deep convolutional neural networks and lstms. *arXiv preprint arXiv:1601.05610*, 2016.

[13] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[14] B. Lorch, S. Agarwal, and H. Farid. Forensic reconstruction of severely degraded license plates. *Electronic Imaging*, 2019(5):529–1, 2019.

[15] B. Lorch, A. Maier, and C. Riess. Reliable jpeg forensics via model uncertainty. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2020.

[16] A. Maier, B. Lorch, and C. Riess. Toward reliable models for authenticating multimedia content: Detecting resampling artifacts with bayesian neural networks. In *International Conference on Image Processing*, pages 1251–1255. IEEE, 2020.

[17] R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, R. Cipolla, and A. Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc., 2017.

[18] D. Moussa, A. Maier, F. Schirrmacher, and C. Riess. Sequence-based recognition of license plates with severe out-of-distribution degradations. In *International Conference on Computer Analysis of Images and Patterns*, pages 175–185. Springer, 2021.

[19] S. Qiao, Y. Zhu, X. Li, T. Liu, and B. Zhang. Research of improving the accuracy of license plate character segmentation. In *International Conference on Frontier of Computer Science and Technology*, pages 489–493. IEEE, 2010.

[20] S. Rasheed, A. Naeem, and O. Ishaq. Automated number plate recognition using hough lines and template matching. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, pages 24–26, 2012.

[21] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[22] G. Rossi, M. Fontani, and S. Milani. Neural network for denoising and reading degraded license plates. In *International Conference on Pattern Recognition*, pages 484–499. Springer, 2021.

[23] S. M. Silva and C. R. Jung. License plate detection and recognition in unconstrained scenarios. In *Proceedings of the European Conference on Computer Vision*, pages 580–596, 2018.

[24] J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík, and P. Zemčík. Holistic recognition of low quality license plates by cnn using track annotated data. In *International Conference on Advanced Video and Signal Based Surveillance*, pages 1–6. IEEE, 2017.

[25] A. Spruck, M. Hawesch, A. Maier, C. Rieß, J. Seiler, and A. Kaup. 3D Rendering Framework for Data Augmentation in Optical Character Recognition. In *International Symposium on Signals, Circuits and Systems*, 2021.

[26] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi. An algorithm for license plate recognition applied to intelligent transportation system. *Transactions on Intelligent Transportation Systems*, 12(3):830–845, 2011.

[27] Y. Wen, P. Vicol, J. Ba, D. Tran, and R. Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *International Conference on Learning Representations*, 2018.

[28] Y. Zou, Y. Zhang, J. Yan, X. Jiang, T. Huang, H. Fan, and Z. Cui. A robust license plate recognition model based on bi-lstm. *IEEE Access*, 8:211630–211641, 2020.