Less is More: Reversible Steganography with Uncertainty-Aware Predictive Analytics

Ching-Chun Chang, Xu Wang, Sisheng Chen and Isao Echizen

Artificial neural networks have advanced the frontiers of reversible steganography. The core strength of neural networks is the ability to render accurate predictions for a bewildering variety of data. Residual modulation is recognised as the most advanced reversible steganographic algorithm for digital images and the pivot of which is the predictive module. The function of this module is to predict pixel intensity given some pixel-wise contextual information. This task can be perceived as a low-level vision problem and hence neural networks for addressing a similar class of problems can be deployed. On top of the prior art, this paper analyses the predictive uncertainty and endows the predictive module with the option to abstain when encountering a high level of uncertainty. Uncertainty analysis can be formulated as a pixel-level binary classification problem and tackled by both supervised and unsupervised learning. In contrast to handcrafted statistical analytics, learning-based analytics can learn to follow some general statistical principles and simultaneously adapt to a specific predictor. Experimental results show that steganographic performance can be remarkably improved by adaptively filtering out the unpredictable regions with the learning-based uncertainty analysers.

Index Terms—Deep learning, neural networks, predictive analytics, reversible steganography, uncertainty quantification.

I. INTRODUCTION

REVERSIBLE steganography is the practice of hiding information into multimedia signals subject to removable distortion [1]. It is an essential technique for data authentication [2]–[9]. In this age of automation and big-data analytics, the volume and variety of data from websites, social networks, electronic devices, and physical sensors are growing rapidly. The demand for maintaining data integrity and verifying data origin has also soared with the growth of data traffic. To cope with communications over insecure channels, authentication messages (e.g. cryptographic signatures, device fingerprints, and timestamps) can be embedded into digital media via this reversible mechanism. The adoption of reversible steganography as an authentication tool can prevent the accumulation of distortion over a series of data transactions.

A distortion-free authentication mechanism would boost many applications. The ongoing global pandemic of coronavirus disease has accelerated digital transformation, and a flood of data is being garnered and analysed with real-time epidemiological surveillance tools [10]. Amidst the crisis, healthcare sector has been struck by a surge in cybercrime [11]. Malicious attacks such as data poisoning and adversarial perturbations can distort the tracking of disease outbreaks and corrupt the data-centric healthcare systems [12]–[14]. To mitigate such threats, a distortion-free authentication mechanism is needed to identify authorised data providers as well as to ensure data accuracy and consistency.

Advances in deep learning have led to significant breakthroughs in many branches of science [15]. Steganography with deep learning has emerged as a promising research paradigm in which deep neural networks (DNNs) are used to build state-of-the-art steganographic systems [16]–[23]. Current DNN-based methods for reversible steganography can be categorised into either end-to-end or modular pipelines. Typically, the end-to-end pipeline trains a pair of encoder and decoder networks jointly to simulate a reversible steganographic system [24]. This encoder-decoder architecture provides an efficient pathway for neural networks to learn to embed/extract confidential messages into/from carrier signals. While the monolithic end-to-end structure often offers high steganographic capacity, the intricate reversing mechanism is difficult, if not impossible, to be learnt by using the existing neural network models, resulting in irrecoverable distortion in the images. One way to overcome the problem of imperfect reversibility is through modularisation. The modular pipeline deploys neural networks to analyse the data distribution and handles reversibility through established reversible steganographic techniques [25].

Residual modulation is among the most advanced reversible steganographic techniques for digital images [29]–[30]. The residual modulation scheme is comprised of a module that predicts pixel intensity on the basis of some pixel-wise contextual information and a module that encodes/decodes a message bitstream into/from residuals (i.e. prediction errors). A more accurate predictor leads to a better rate-distortion trade-off. There are basically two approaches to programming predictive algorithms. The rule-based approachformulates a predictive model with handcrafted or fixed parameters. The model typically has low computational complexity, so it can be implemented with a minimal number of operations and parameters. The learning-based approach finds optimal parameters for a predictive model by feeding it with vast amounts of data. A DNN model usually has millions of learnable parameters in order to deal with a bewildering range of input images and extract high-level
Abstract features for making predictions \cite{31}--\cite{33}. A recent study has demonstrated that a convolutional neural network (CNN) with multi-scale receptive field can extract multi-scale image features and hence has better performance than that of traditional predictors, which use relatively limited contextual information \cite{34}. Another study has established a connection between low-level computer vision and intensity prediction \cite{35}, and adopts the memory-persistent network (MemNet) originally proposed for super-resolution imaging and image denoising \cite{36}.

Prediction accuracy is content-dependent: in general, it is high for smooth regions and low for rough regions. A potential way to improve prediction accuracy is to adaptively filter out some unpredictable regions. This can be done by formulating a function that analyses pixel correlations and estimates local pattern complexity \cite{37}--\cite{39}. However, the identified unpredictable regions are bounded within the confines of general principles (e.g. smoothness prior) laid out by handcrafted statistical analytics. In addition, we argue that this kind of prescriptive method appears to fall short of capturing true uncertainty (or predictability), defined as the degree to which a correct prediction can be made by a given predictor. From our perspective, uncertainty is not only associated with the image content but also with the predictor used. We thus must analyse for which types of patterns a given predictor has the highest probability of making accurate predictions. Learning-based analytics can learn to follow some of the basic principles and, at the same time, adapt to a specific predictor because it does not have a preconceived notion.

The concept of self-awareness, albeit illusive and indefinite, can be understood as the ability to know one’s own strengths and limitations. While there is no indication that contemporary artificial intelligence is anywhere close to engendering self-awareness, a rudimentary kind of self-aware machine should be able to realise whether or not it can handle a given query: the machine knows what it knows and does not know. It thus could be given a reject option and be allowed to abstain from making a prediction when there is a large amount of uncertainty \cite{40}--\cite{44}. Following the ‘less is more’ philosophy, our objective is to build an uncertainty-aware predictive module by endowing it with an option to exercise abstention and thereby improve prediction accuracy.

In this paper, we study uncertainty-aware predictive analytics for reversible steganography. In summary, our aim is to create a robust and reliable predictive module with a reject option. To this end, we introduce both supervised and unsupervised learning for training uncertainty analysers by which each pixel in an image is classified into either predictable or unpredictable. For supervised learning, we construct a ground-truth dataset by representing uncertainty as quantised residuals and train a neural network to segment predictable and unpredictable regions. For unsupervised learning, we formulate an ad hoc loss function and apply the concept of transfer learning to estimate the uncertainty on the basis of a pre-trained neural network. The uncertainty analysers enable selective prediction and adaptive residual modulation, thereby improving the steganographic rate-distortion performance. We carry out an ablation study to clarify the contribution of the learning-based uncertainty analysers to the steganographic system, and compare their performance with that of a representative handcrafted statistical analyser.
The remainder of this paper is organised as follows. Section II presents the proposed uncertainty analysis methods. Section III evaluates the effects of the proposed methods in comparison to the benchmarks. Concluding remarks are given in Section IV.

II. METHODOLOGY

We start with an outline of the encoding/decoding framework, followed by a detailed description of the steganographic mechanisms. Then, we present a neural network model for pixel intensity prediction, which leads to the notion of uncertainty-aware machine learning. To enable self-adaptive prediction, both supervised and unsupervised learning frameworks are introduced. Furthermore, a variance-based complexity analysis method is established as a benchmark to which our proposed uncertainty analysis methods are compared.

A. Framework Overview

The objective of reversible steganography is to modify a cover image \( x \) in an imperceptible and reversible manner to embed a message \( m \), resulting in a stego image \( x' \). A reversible steganographic system can be divided into an encoding phase (message embedding) and a decoding phase (message extraction and image restoration), as outlined in Figure 1. The encoding is operated as follows. To begin with, the image is pre-processed to prevent pixel intensity overflow. This pre-processing is performed by scaling down near-saturated pixels (i.e. pixels with an intensity value around the maximum or minimum). An overflow-status register \( v \) is used to flag the processed pixels in order to distinguish them from unprocessed ones with the same values. It is required for perfect reversibility, and thus should be concatenated with the intended message as a part of the payload \( \omega \). The image is split into the context and query sets, denoted respectively by \( x_C \) and \( x_Q \), via a pre-defined chequerboard lattice. Predictions of query pixels (denoted by \( y \)) along with an uncertainty map (denoted by \( \phi \)) are derived from the context pixels by applying two separate neural networks. The predictable pixels are identified in accordance to the uncertainty map and then selected to carry information. The residuals between the adaptively selected pixels and their predicted counterpart, as denoted by \( \epsilon \) are computed, and the payload is embedded by modulating the residuals. The selected query pixels are modified by adding the modulated residuals to the predicted intensities. The context and query pixels are then merged together to obtain a stego image. The decoding is operated in a similar way. It begins by splitting the stego image into the context and query sets. Since the context set remains unmodified during the encoding phase, the same predictions as well as an identical uncertainty map can be derived. Then, the predictable pixels are identified, the residuals are calculated and de-modulated, the payload is extracted, and the finally pixels are restored. Post-processing is required to segment the payload into the message and overflow-status register. The latter is then used to scale up those near-saturated pixels to their original intensities, yielding a distortion-free image. A list of variables and operations is provided in Table I and a concise summary of encoding and decoding phases is shown in Algorithms 1 and 2.

B. Algorithmic Details

Let us denote by \( \alpha \) a stego-channel parameter that regulates how a residual is modulated, where \( \alpha \in \mathbb{Z}^+ \). It acts as a threshold value between the carrier and non-carrier residuals, such that

\[
\epsilon \in \begin{cases} 
\text{carrier} & \text{if } \text{abs}(\epsilon) < \alpha, \\
\text{non-carrier} & \text{otherwise},
\end{cases}
\]

where \( \text{abs}(\epsilon) \) denotes the absolute magnitude of a residual. According to the law of error, the frequency of a residual can be expressed as an exponential function of its magnitude and therefore follows a zero-mean Laplacian distribution \([45]\). In other words, the residuals with magnitude 0 should occur most frequently, followed by \( \pm 1, \pm 2 \), and so forth. To embed a message, we modulate a residual symmetrically from magnitude 0 to \( \pm 1 \) or keep it unchanged, giving three different

---

**Algorithm 1: Encoding**

**Input:** \( x \) and \( m \)  
**Output:** \( x' \)

// pre-processing  
- \([\bar{x}, v] = \text{scale}_v(x \mid \alpha)\)  
- \(\omega = \text{concat}(m, v)\)

// encoding  
- \([x_C, x_Q] = \text{split}(\bar{x})\)  
- \(y = \text{net}_y(x_C)\)  
- \(\phi = \text{net}_\phi(x_C)\)  
- \([\tilde{x}_Q, \tilde{y}] = \text{select}(x_Q, y \mid \phi)\)  
- \(\epsilon = \bar{x} - \tilde{y}\)  
- \(\epsilon' = \text{modulate}(\epsilon, \omega \mid \alpha)\)  
- \(\bar{x}_Q = \tilde{y} + \epsilon'\)  
- \(x'_Q = \text{replace}(x_Q, \bar{x}_Q \mid \phi)\)  
- \(x' = \text{merge}(x_C, x'_Q)\)

---

**Algorithm 2: Decoding**

**Input:** \( x' \)  
**Output:** \( x \) and \( m \)

// decoding  
- \([x_C, x'_Q] = \text{split}(x')\)  
- \(y = \text{net}_y(x_C)\)  
- \(\phi = \text{net}_\phi(x_C)\)  
- \([\tilde{x}'_Q, \tilde{y}] = \text{select}(x'_Q, y \mid \phi)\)  
- \(\epsilon' = x'_Q - \tilde{y}\)  
- \([\epsilon, \omega] = \text{de-modulate}(\epsilon' \mid \alpha)\)  
- \(x_Q = \tilde{y} + \epsilon\)  
- \(x_Q = \text{replace}(x'_Q, \tilde{x}_Q \mid \phi)\)  
- \(x = \text{merge}(x_C, x_Q)\)

// post-processing  
- \([m, v] = \text{de-concat}(\omega)\)  
- \(x = \text{scale}_v(x, v \mid \alpha)\)
states, thereby enabling one ternary digit ($\log_2 3 \approx 1.58$ bits of information) to be represented. To circumvent the inherent problem with numerical base conversion, we embed one or two bits with a probability of 0.5 each in the following way:

$$\{ \varepsilon' | \varepsilon = 0 \} = \begin{cases} 0 & \text{if } \omega_t = 0, \\ -1 & \text{if } \omega_t, \omega_{t+1} = [1, 0], \\ +1 & \text{if } \omega_t, \omega_{t+1} = [1, 1], \end{cases}$$

where $\omega_t$ denotes the $t$th payload bit. This compromise solution enables the embedding of 1.5 bits on average. The absolute magnitude of each carrier residual other than those with magnitude 0 is either kept unchanged or increased by 1 to embed one bit of information; that is,

$$\{ \varepsilon' | 0 < \text{abs}(\varepsilon) < \alpha \} = \begin{cases} 2\varepsilon & \text{if } \omega_t = 0, \\ 2\varepsilon + \text{sgn}(\varepsilon) \cdot 1 & \text{if } \omega_t = 1, \end{cases}$$

where $\text{sgn}(\varepsilon)$ denotes the sign of a residual (either positive or negative). For each non-carrier residual, we increase its absolute magnitude by $\alpha$ to separate it from other modulated residuals; that is,

$$\{ \varepsilon' | \text{abs}(\varepsilon) \geq \alpha \} = \varepsilon + \text{sgn}(\varepsilon) \cdot \alpha.$$

The modulation in the residual domain produces a pro rata increase or decrease in the spatial domain. Let $x_{i,j}$ denote a pixel at co-ordinates $(i, j)$. To prevent overflow, we scale down near-saturated pixels (determined by $\alpha$) in advance:

$$x_{i,j} = \begin{cases} x_{i,j} + \alpha & \text{if } x_{i,j} \in [x_{\text{min}}, x_{\text{min}} + \alpha - 1], \\ x_{i,j} - \alpha & \text{if } x_{i,j} \in [x_{\text{max}} - \alpha + 1, x_{\text{max}}], \end{cases}$$

where $x_{\text{min}}$ and $x_{\text{max}}$ denote the minimum and maximum of all possible pixel values, respectively (e.g. values 0 and 255 for 8-bit colour depth). The downscaling causes collisions between the scaled and unscaled pixel values. Such collisions are prevented by setting flags to 1 (true) for the scaled pixels and to 0 (false) for the unscaled pixels.

### C. Predictive Module

The aim of intensity prediction is to extrapolate the intensities of query pixels from the observed context pixels. A common way to sample the context and query pixels is by using a chequerboard-like lattice, as illustrated in Figure 2.

The lattice forms a uniform sampling pattern such that each query pixel is surrounded by four context pixels connected horizontally and vertically. This particular prediction task can be viewed as a low-level vision problem that concerns the spatial correlations as well as low-level image features (e.g. colour, texture, and shape). This perception enables intensity prediction to be handled with neural network models designed for solving a similar class of problems such as super-resolution reconstruction, missing-data restoration, and noise reduction. The residual dense network (RDN) is considered state of the art for such low-level vision problems [46]. The RDN model is characterised by its network-in-network architecture comprising a tangled labyrinth of residual and dense connections, as illustrated in Figure 3. It has a hierarchical structure that extracts image features at both local and global levels. At the local level, a number of $3 \times 3$ convolutional layers are densely connected with a non-linear ReLU activation function applied after each layer [47]. The outputs from previous layers are then integrated by a $1 \times 1$ convolutional layer that fuses multiple features in accordance with their saliencies [48]. A skip connection between the input and output of this residual dense block (RDB) is established to mitigate the problem of vanishing gradients [49]. At the global level, the input passes through a number of RDBs along and a few intermediate convolutional layers with a skip connection between a shallow layer and a deep layer. In implementation, the number of RDBs is specified to 3 and the number of convolutional layers in each RDB is configured to 5. The RDN is trained to minimise the $\ell_1$ norm via the back-propagation algorithm [50]. There are two basic ways to initialise the query pixels of the input images: zero initialisation and local-mean initialisation. As the results are virtually the same, we use the former for simplicity.

Although neural networks are capable of making accurate predictions on average, the predicted intensities of particular pixels can be heavily skewed. Identifying and filtering out the unpredictable pixels should further increase the overall accuracy. These singular pixels often appear in some highly textured areas. While it is possible to capture this content dependent nature by using a pattern analysis algorithm, we argue that the identified complex regions are not completely equivalent to the desired unpredictable regions. We thus pro-
pose estimating the uncertainty of each prediction directly. Uncertainty analysis can impart a rudimentary kind of self-awareness to the predictive module in the sense that the module is given an option to abstain from making predictions when the confidence level is low. In the following, we introduce both supervised and unsupervised learning for uncertainty analysis.

D. Uncertainty Analysis with Supervised Learning

Let us consider a simplified case in which each pixel is classified as either predictable or unpredictable, denoted by 1 and 0 respectively. This is a pixel-level binary classification problem, which can also be viewed as a special kind of image segmentation problems. This means we can employ a neural network model originally developed for image segmentation to deal with uncertainty analysis. The remaining question is how to construct the ground-truth targets to which the model is trained to fit. To enable supervised learning, we create the ground-truth targets by quantising the prediction residuals to 1 if their magnitudes are less than a given threshold and to 0 otherwise. In practice, we set this threshold to \( \alpha \) to link the predictable pixels with the carrier pixels. Since residuals can only be computed for query pixels, we approximate the residuals for context pixels by linear interpolation—that is, the mean of four neighbours. A diagram of the ground-truth generation process is shown in Figure 4. In the implementation, we deploy the classic U-shaped network (U-Net) widely used for image segmentation [51]. This model comprises a pair of encoder and decoder with skip connections between mirrored layers. The image passes from top to bottom through consecutive downscaling convolutions, and is then updated through a bottom-up pass with a succession of upscaling convolutions. The model learns to assemble a precise output on the basis of multi-resolution features. A sigmoid activation function is placed at the end of the model to bound the output values between 0 and 1. The U-Net is trained to minimise the cross-entropy loss function:

\[
\mathcal{L}_\phi = -\sum_i \sum_j \psi_{i,j} \log(\phi_{i,j}) + (1 - \psi_{i,j}) \log(1 - \phi_{i,j}),
\]

where \( \psi_{i,j} \) is the ground truth for \( \phi_{i,j} \).

E. Uncertainty Analysis with Unsupervised Learning

Learning to generate uncertainty maps in a supervised fashion, albeit feasible, involves extensive computational time and memory for constructing a ground-truth dataset. It is possible to analyse uncertainty in a partially unsupervised manner. The idea is connected to transfer learning—we employ a pre-trained predictive model and formulate an ad hoc loss function. Let us denote by \( x \) the ground-truth image, \( y \) the image predicted by the pre-trained RDN, and \( \phi \) the uncertainty map to be learnt. We append an additional trainable convolutional layer along with a sigmoid activation function prior to the last layer of the pre-trained neural network, which has learnt to extract rich information from the input image. This dual-branch neural network, as illustrated in Figure 5, is trained to minimise the following loss function:

\[
\mathcal{L}_\phi = C(x, y \mid \phi) + \lambda R(\phi),
\]

where \( C \) is the cost function and \( R \) is the regularisation term weighted by \( \lambda \). Each pixel of the output map, prior to binarisation, is a real number in \([0, 1]\) due to the sigmoid function. A larger value of \( \phi_{i,j} \) indicates a higher confidence in predicting the corresponding pixel. The cost function is defined by

\[
C(x, y \mid \phi) = \frac{\| (x - y) \odot \phi \|_1}{\sum_i \sum_j \phi_{i,j}},
\]

where \( \| \cdot \|_1 \) denotes the \( \ell_1 \) norm and \( \odot \) denotes the Hadamard (element-wise) product. This function is designed to reflect the distance between the actual and predicted images over the selected area, normalised by the size of the area. The regularisation term is used to penalise the model for abstention and thus encourage it to reveal more predictable regions, as defined by

\[
R(\phi) = \sum_i \sum_j (1 - \phi_{i,j})^2.
\]

The weight for balancing between two loss terms is determined empirically.

F. Complexity Analysis with Local Variance

This study make comparisons with a traditional statistical analysis for benchmarking purpose to validate that the learning-based analysis is competitive. Common statistical methods for local complexity estimation can be considered as variations of a variance-based analysis [37]–[39], [52].

![Fig. 4. Supervised learning for intensity prediction and uncertainty estimation with ground-truth generation.](image)

![Fig. 5. Unsupervised learning for uncertainty analysis via a pre-trained neural network with an additional trainable layer.](image)
This analysis model computes local variance of an image and then quantises the variance values to different levels of local complexity. It performs variance filtering in such a way that the value of each output pixel is the variance of the 4-connected neighbourhood around the corresponding input pixel:

$$\sigma_{i,j} = \frac{1}{N} \sum_{k=1}^{N} \left(\text{hood}_k(x_{i,j}) - \mu_{i,j}\right)^2,$$

where $\text{hood}_k(x_{i,j})$ denotes an element of the pixel neighbourhood, $N$ the number of neighbours, and $\mu_{i,j}$ the mean of the neighbours. The number of neighbouring pixels is 4 in accordance with the defined pixel connectivity, but can be less than 4 for pixels on the borders if border padding is not applied. A small local variance indicates a low local complexity, so the variance map can be binarised to a complexity map with a threshold $\theta$, as given by

$$\phi_{i,j} = \begin{cases} 1 & \text{if } \sigma_{i,j} < \theta, \\ 0 & \text{otherwise}. \end{cases}$$

The choice of the threshold value affects the boundary between the predictable and unpredictable areas. An example of binarising a variance map with different threshold values is shown in Figure 6. The threshold value is determined in accordance with the proportion of predictable pixels calculated from the uncertainty map. In other words, the threshold value is determined by the percentile below which a given percentage of area is classified as predictable, as illustrated in Figure 7. This makes the baseline statistical model comparable to the neural network model as their output maps contain an identical percentage of predictable pixels.

### III. Experiments

The objective of our experiments is to identify the improvements delivered by uncertainty analysis for several aspects of reversible steganography. We carry out an ablation study to assess the contribution of the learning-based methods benchmarked against a variance-based method. We begin by describing the datasets and benchmarks. Then, we visualise the results produced by different methods. A comparison of the classification accuracy among different methods is provided. Furthermore, as the primary concern for residual modulation algorithm, the residual distribution is closely examined with different statistical measurements for concentration. Finally, we evaluate the steganographic rate-distortion performance to confirm the improvements.

#### A. Datasets and Benchmarks

The neural networks were trained and tested on the BOSSbase dataset [53]. This dataset was developed from an academic competition for digital steganography and has a collection 10,000 greyscale photographs covering a wide diversity...
of subjects, with a few samples displayed in Figure 8. The training/test split ratio was 80/20. The inference set was composed of standard test images from the USC-SIPI dataset [54], as displayed in Figure 9. A comparative study was carried out to assess the supervised learning (SL) and unsupervised learning (UL) methods, benchmarked against a local variance (LV) method [37]–[39], [52]. Apart from the RDN, the prior art predictive neural networks (MS-CNN [34] and MemNet [35]) were also used as baseline models to evaluate the rate-distortion performance. The steganographic parameter was set to $\alpha = 2$ in all experiments.

**B. Visual Comparison**

Figure 10 visualises the residuals between the actual and predicted images, the ground-truth uncertainty maps used for supervised learning, and the processed ground-truth uncertainty maps used for visual comparison. The ground-truth maps are produced by thresholding the absolute residuals with $\alpha$. The processed maps are obtained by setting the intensities of the context pixels to zero in order to emphasise the query pixels classified as predictable. The numerical data for the former describes the percentage of predictable regions over the whole image, whereas the numerical data for the latter denotes the percentage over the query pixels (i.e. half size of the image) only. Figures 11 and 12 display respectively the visual results for supervised and unsupervised learning; the outputs from the neural networks, the uncertainty maps obtained through binarisation with a threshold of 0.5, and the misclassified areas identified by matching against the ground truth. The raw outputs from the supervised method closely resembles the ground truth in comparison to those learnt by the unsupervised counterpart. It seems that the supervised method is able to learn relatively fine-grained patterns. At first glance, the unsupervised method seems to suffer from a problem of checkerboard-like artefacts if we observe its raw outputs. Nonetheless, after we binarise the outputs and
discard the results for the context pixels, the uncertainty maps produced in an unsupervised manner are aligned with those generated in a supervised manner, although the percentage of predictable regions varies from image to image. It can also be observed that the supervised analyser achieves a lower misclassification rate per query pixel than the unsupervised one. For both analysers, the predictable pixels are largely concentrated in smooth areas, indicating that the learning-based analysers are capable of capturing some of the basic statistical principles. The visual results match our hypothesis that the predictability is associated with the image content. Figure 13 shows the variance maps generated by using a local variance filter. Figures 14 and 15 demonstrates the complexity maps obtained by configuring $\theta$ in such a way that the percentage of predictable pixels in each complexity map is identical to the percentage in the corresponding uncertainty map. We denote by $LV(\theta \sim SL)$ a variance-based complexity analyser where $\theta$ is determined adaptively in order to make it comparable to the supervised uncertainty analyser, and by $LV(\theta \sim UL)$ likewise. While the misclassification rate of the unsupervised analyser is on par with the variance-based analyser, the supervised analyser has a lower misclassification rate than the variance-based analyser.

C. Classification Performance

We use the receiver operating characteristic (ROC) curve to further examine the diagnostic ability of different analysers on the inference set, as plotted in Figure 16. The curves show the true positive rate against the false positive rate at various threshold values. We consider each image as a a classification dataset and each pixel as a sample to be classified. The curves are generated by adjusting the binarisation threshold for the real-valued maps (i.e. raw outputs from the neural networks and variance maps). The greater the deviation from the diagonal line (line of no-discrimination), the better the classification performance. The performance is evaluated numerically by the area under the curve (AUC). It can be seen that the supervised analyser rises above others, and that the variance-based analyser outdoes the unsupervised analyser slightly. Figure 17 depicts the classification accuracy in a large-scale assessment on the test set using a five-number summary: the minimum, the first quartile, the median, the third quartile, and the maximum. Overall, the supervised analyser demonstrates a distinct advantage in classification performance at the cost of building a ground-truth dataset.

D. Residual Distribution

The residual modulation algorithm is designed based on the observation that residuals normally follow a Laplacian distribution and that a more accurate predictive model yields a more concentrated distribution. The residual distribution is a determining factor affecting steganographic rate-distortion performance. Figure 18 describes the residual distribution of each individual image by the probability distribution function (PDF) labelled with the distribution’s variance as a measurement for concentration. Figure 19 visualises the degree of dispersion and skewness in the variance values derived from the test set. Figure 20 describes the residual distribution of each individual image by the cumulative distribution function (CDF) labelled with the distribution’s 95th percentile as another measurement for concentration. Figure 21 visualises the statistical population of the percentile values evaluated on the test set. A smaller value of variance indicates a more concentrated distribution and a smaller value of percentile suggests likewise. We can observe that the supervised learning is superior to the unsupervised learning in every concentration measurement. The supervised method outdistances its variance-based counterpart, whereas the unsupervised method is at a similar level as its variance-based counterpart. The trend for the large-scale assessment are similar to that for the individual images.

E. Rate-Distortion Performance

The results for steganographic rate-distortion performance are shown in Figure 22. We evaluate the steganographic distortion on the basis of the peak signal-to-noise ratio (PSNR) (expressed in decibels; dB) and the steganographic capacity on the basis of the embedding rate (expressed in bits per pixel; bpp). Overall, the image quality decreases steadily as the embedding rate increases. In addition to this, although the provision of the abstention option limits the maximum capacity, it results in a better trade-off between the rate and distortion. The RDN with a supervised analyser is ranked in the top tier followed by its variance-based counterpart. The RDN with an unsupervised analyser and its variance-based counterpart are ranked in the second tier. The bottom tier
can observe a significant improvement over the state-of-the-art methods with and without an abstention option. We could be introduced by modifying the network architecture MemNet, and MS-CNN. While the improvement offered by methods, ranked in descending order, are the plain RDN, areas labelled with error rate.

![Fig. 15. Visual results for LV from the end-to-end paradigm. We look forward to the future from the modular paradigm, new development may also come as a regression problem in a more general sense. Apart from the modular paradigm, new development may also come from the end-to-end paradigm. We look forward to the future progress on reversible steganography with deep learning.](image)

methods, ranked in descending order, are the plain RDN, MemNet, and MS-CNN. While the improvement offered by the unsupervised analyser seems modest, potential refinements could be introduced by modifying the network architecture and the loss function. There is a notable performance gap between methods with and without an abstention option. We can observe a significant improvement over the state-of-the-art predictors in terms of the rate-distortion performance. The results confirm the validity of both supervised and unsupervised frameworks for uncertainty analysis.

IV. CONCLUSION

Although reversible steganography with deep learning is still in its infancy, systems that transform some of their modules into neural networks have already been proven to outperform their original version. In this paper, we investigate learning-based uncertainty analysers for excluding the unpredictable pixels from the reversible steganographic process. We propose both supervised and unsupervised learning frameworks for uncertainty analysis. The experimental results demonstrate that neural networks are capable of learning local uncertainty and consequently able to decide whether to predict or abstain. We have shown a substantial improvement over predictors without an uncertainty analyser as well as predictors with a statistical analyser. The supervised and unsupervised methods could both be further improved by incorporating state-of-the-art neural network models and by rethinking the loss functions. While we consider uncertainty analysis to be a pixel-level binary classification problem, it can be formulated as a regression problem in a more general sense. Apart from the modular paradigm, new development may also come from the end-to-end paradigm. We look forward to the future progress on reversible steganography with deep learning.

REFERENCES

[1] Y.-Q. Shi, X. Li, X. Zhang, H. Wu, and B. Ma, “Reversible data hiding: Advances in the past two decades,” *IEEE Access*, vol. 4, pp. 3210–3237, 2016.

[2] J. Fridrich, M. Goljan, and R. Du, “Lossless data embedding—New paradigm in digital watermarking,” *EURASIP J. Adv. Signal Process.*, vol. 986842, no. 2002, pp. 185–196, 2002.

[3] C. De Vleeschouwer, J.-F. Delaigle, and B. Macq, “Circular interpretation of bijective transformations in lossless watermarking for media asset management,” *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 97–105, 2003.

[4] J. Tian, “Reversible data embedding using a difference expansion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, 2003.

[5] M. U. Celik, G. Sharma, and A. M. Tekalp, “Lossless watermarking for image authentication: a new framework and an implementation,” *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 1042–1049, 2006.

[6] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, 2006.

[7] S. Lee, C. D. Yoo, and T. Kalker, “Reversible image watermarking based on integer-to-integer wavelet transform,” *IEEE Trans. Inf. Forensics Secur.*, vol. 2, no. 3, pp. 321–330, 2007.

[8] D. Colucc, “Low distortion transform for reversible watermarking,” *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 412–417, 2012.

[9] J. Wang, J. Ni, X. Zhang, and Y.-Q. Shi, “Rate and distortion optimization for reversible data hiding using multiple histogram shifting,” *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 315–326, 2017.

[10] N. Collier, S. Doan, A. Kazawoe, R. M. Goodwin, M. Conway, T. Tateno, Q.-H. Ngo, D. Dien, A. Kawtrakul, K. Takeuchi, M. Shigematsu, and K. Taniguchi, “BioCaster: Detecting public health rumors with a Web-based text mining system,” *Bioinformatics*, vol. 24, no. 24, pp. 2940–2941, 2008.

[11] H. S. Lallie, L. A. Shepherd, J. R. C. Nurse, A. Erola, G. Epiphaniou, C. Maple, and X. Bekaletis, “Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic,” *Comput. & Secur.*, vol. 105, pp. 102 248:1–20, 2021.

[12] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proc. Int. Conf. Learn. Representations (ICLR)*, San Diego, CA, USA, 2015, pp. 1–11.

[13] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2574–2582.

[14] L. Muñoz González, B. Biggio, A. Demontis, A. Paudice, V. Wongrasamee, E. C. Lupu, and F. Roli, “Towards poisoning of deep learning algorithms with back-gradient optimization,” in *Proc. ACM Workshop Artif. Intell. Secur. (AISec)*, Dallas, TX, USA, 2017, pp. 27–38.

[15] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning.” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[16] S. Baluja, “Hiding images in plain sight: Deep steganography,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 2066–2076.

[17] J. Hayes and G. Danezis, “Generating steganographic images via adversarial training,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 1951–1960.

[18] W. Tang, S. Tan, B. Li, and J. Huang, “Automatic steganographic distortion learning using a generative adversarial network,” *IEEE Signal Process. Lett.*, vol. 24, no. 10, pp. 1547–1551, 2017.

[19] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, “HiDDen: Hiding data with deep networks,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 682–697.

[20] D. Volkonskiy, I. Nazarov, and E. Burnaev, “Steganographic generative adversarial networks,” in *Proc. Int. Conf. Mach. Vis. (ICMV)*, vol. 11433, Amsterdam, Netherlands, 2019, pp. 1–15.

[21] E. Wengrowski and K. Dana, “Light field messaging with deep photorealistic steganography,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 1515–1524.

[22] M. Tancik, B. Mildenhall, and R. Ng, “StegaStamp: Invisible hyperlinks in physical photographs,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 2114–2123.

[23] X. Luo, R. Zhan, H. Chang, F. Yang, and P. Milanfar, “Distortion agnostic deep watermarking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 13 543–1545.

[24] X. Dai, K. Bai, B. Li, D. Guo, E. Zhang, and C. Qin, “Reversible image steganography scheme based on a U-Net structure,” *IEEE Access*, vol. 7, pp. 9314–9323, 2019.

[25] C.-C. Chang, “Adversarial learning for invertible steganography,” *IEEE Access*, vol. 8, pp. 198 425–198 435, 2020.

[26] D. M. Thodi and J. J. Rodriguez, “Expansion embedding techniques for reversible watermarking,” *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, 2007.

[27] M. Fallahpour, “Reversible image data hiding based on gradient adjusted prediction,” *IEICE Electron. Exp.*, vol. 5, no. 20, pp. 870–876, 2008.
Fig. 16. ROC curves with numerical data denoting AUC.

Fig. 17. Classification accuracy evaluated on test set.

Fig. 18. Probability distribution function of residual distribution with numerical data denoting distribution’s variance.

Fig. 19. Variance of residual distribution evaluated on test set.

Fig. 20. Cumulative distribution function of residual distribution with Fig. 21. 95th percentile of residual distribution evaluated on test set. Numerical data denoting distribution’s 95th percentile.
Fig. 22. Steganographic rate-distortion curves benchmarked against predictors with and without abstention option.

[28] X. Li, B. Yang, and T. Zeng, “Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection,” IEEE Trans. Image Process., vol. 20, no. 12, pp. 3524–3533, 2011.

[29] I. Dragoi and D. Culcute, “Local-prediction-based difference expansion reversible watermarking,” IEEE Trans. Image Process., vol. 23, no. 4, pp. 1779–1790, 2014.

[30] H. J. Hwang, S. Kim, and H. J. Kim, “Reversible data hiding using least square predictor via the LASSO,” EURASIP J. Image Video Process., vol. 2016, no. 1, pp. 42: 1–12, 2016.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proceedings of Advances in Neural Information Processing Systems (NeurIPS), vol. 25, 2012, pp. 1097–1105.

[32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Representations (ICLR), San Diego, CA, USA, 2015, pp. 1–14.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Boston, MA, USA, 2015, pp. 1–9.

[34] R. Hu and S. Xiang, “CNN prediction based reversible data hiding,” IEEE Signal Process. Lett., vol. 28, pp. 464–468, 2021.

[35] C.-C. Chang, “Neural reversible steganography with long short-term memory,” Secur. Commun. Netw., vol. 2021, pp. 5580272: 1–14, 2021.

[36] Y. Tai, J. Yang, X. Liu, and C. Xu, “MemNet: A persistent memory network for image restoration,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, 2017, pp. 4549–4557.

[37] V. Sainave, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, “Reversible watermarking algorithm using sorting and prediction,” IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 7, pp. 989–999, 2009.

[38] F. Peng, X. Li, and B. Yang, “Adaptive reversible data hiding scheme based on integer transform,” Signal Process., vol. 92, no. 1, pp. 54–62, 2012.

[39] F. Cao, B. An, H. Yao, and Z. Tang, “Local complexity based adaptive embedding mechanism for reversible data hiding in digital images,” Multimed. Tools. Appl., vol. 78, no. 7, pp. 7911–7926, 2019.

[40] C. De Stefano, C. Sansone, and M. Vento, “To reject or not to reject: That is the question—An answer in case of neural classifiers,” IEEE Trans. Syst. Man Cybern., vol. 30, no. 1, pp. 84–94, 2000.

[41] Y. Geifman and R. El-Yaniv, “SelectiveNet: A deep neural network with an integrated reject option,” in Proc. Int. Conf. Mach. Learn. (ICML), vol. 97, Long Beach, CA, USA, 2019, pp. 2151–2159.

[42] S. Thulasidasan, T. Bhattacharya, J. Bilmes, G. Chennupati, and J. Mohd-Yusof, “Combating label noise in deep learning using abstention,” in Proc. Int. Conf. Mach. Learn. (ICML), vol. 97, Long Beach, CA, USA, 2019, pp. 6234–6243.

[43] B. Z. Du, Q. Guo, Y. Zhao, T. Zhi, Y. Chen, and Z. Xu, “Self-aware neural network systems: A survey and new perspective,” Proc. IEEE, vol. 108, no. 7, pp. 1047–1067, 2020.

[44] B. Kompa, J. Snoek, and A. L. Beam, “Second opinion needed: Communicating uncertainty in medical machine learning,” NPJ Digit. Med., vol. 4, no. 1, pp. 4: 1–6, 2021.

[45] E. B. Wilson, “First and second laws of error,” J. Amer. Statist. Assoc., vol. 18, no. 143, pp. 841–851, 1923.

[46] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA, 2018, pp. 2472–2481.

[47] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in Proc. Int. Conf. Artif. Intell. Statist. (AISTATS), Fort Lauderdale, FL, USA, 2011, pp. 315–323.

[48] M. Liu, Q. Chen, and S. Yan, “Network in network,” in Proc. Int. Conf. Learn. Representations (ICLR), Banff, AB, Canada, 2014, pp. 1–10.

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

[50] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Nature, vol. 323, no. 6088, pp. 533–536, 1986.

[51] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI), Munich, Germany, 2015, pp. 234–241.

[52] X. Li, B. Yang, and T. Zeng, “Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection,” IEEE Trans. Image Process., vol. 20, no. 12, pp. 3524–3533, 2011.

[53] P. Bas, T. Filler, and T. Pevny, “Break our steganographic system: The ins and outs of organizing BOSS,” in Proc. Int. Workshop Inf. Hiding (IH), Prague, Czech Republic, 2011, pp. 59–70.

[54] A. G. Weber, “The USC-SIPI image database: Version 5,” USC Viterbi School of Engineering, Signal and Image Processing Institute, Los Angeles, CA, USA, Tech. Rep. 315, 2006.