

Uncertainty Estimation in Image Classification using Monte Carlo Dropout and Variational Inference with BCNN

...

Rita Ortese and Arahí Fernández Monagas

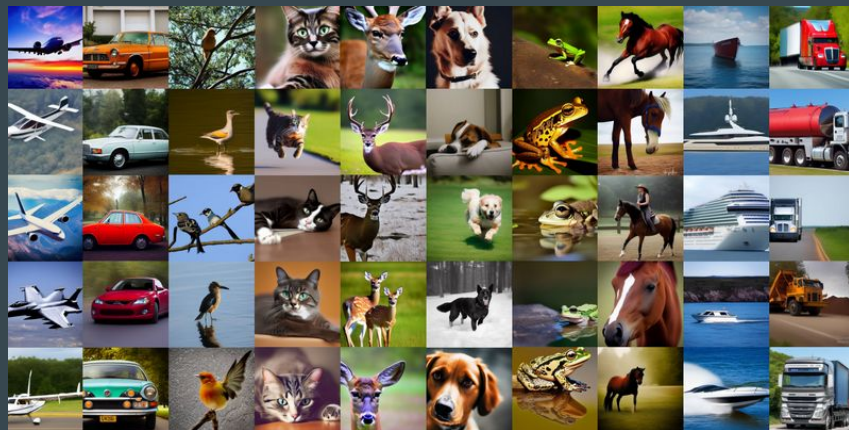
GOAL: Classify an image as a Real or AI



DATASET

We are using **CIFAKE: Real and AI-Generated Synthetic Images**

- It consists of **60,000 synthetically-generated** images and **60,000 real** images, categorized into 2 classes: **REAL** (class 1) and **FAKE** (class 0).
- We split the data into:
 - **100,000** for training (50k per class)
 - **20,000** for testing (10k per class)

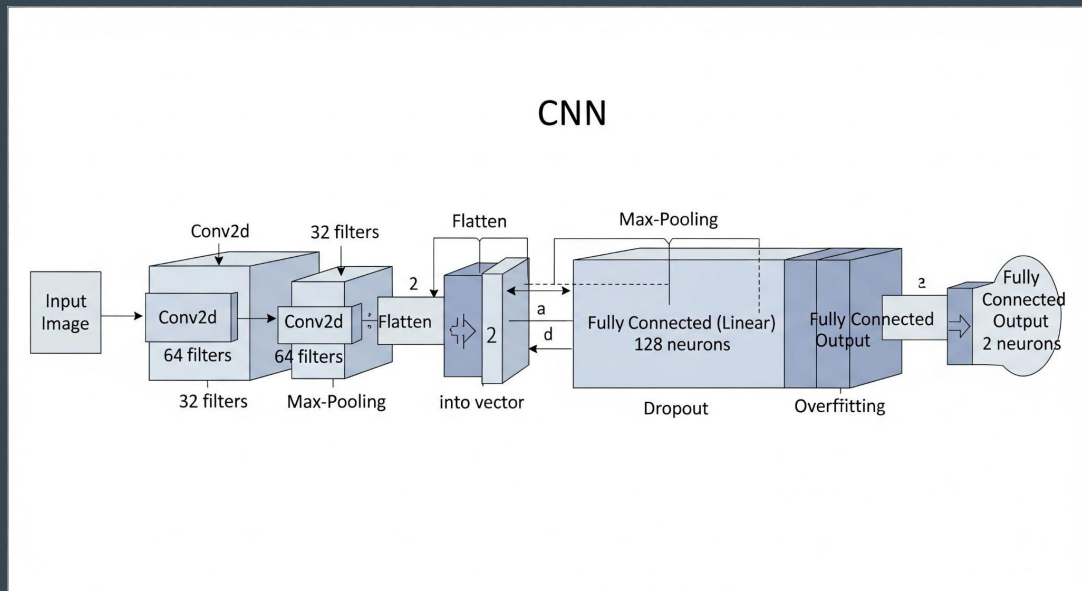


Model 1: MonteCarlo Dropout

Architecture

Model Architecture

- Conv Block 1: conv1
 - (32 filters, 3x3) + MaxPool.
- Conv Block 2: conv2
 - (64 filters, 3x3) + MaxPool.
- Dense Layer:
 - Flatten -> fc1 (128 neurons).
- Dropout (p=0.5):
 - Key layer.
 - Kept active during inference for MC Dropout.
- Output Layer: fc2 for binary classification (REAL / FAKE).



Monte Carlo Dropout (MC-Dropout) for Uncertainty Estimation

What is it?

- The key is to keep the Dropout layer active during the prediction (inference) phase, not just during training.
- Dropout **randomly deactivates** different neurons each time, we get a distribution of different results for the same image.

$$\text{Loss} = - \sum_{i=1}^C y_i \log(\hat{y}_i)$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

$$\mu = \frac{1}{T} \sum_{t=1}^T \hat{y}^{(t)}, \quad \sigma^2 = \frac{1}{T} \sum_{t=1}^T \hat{y}^{(t)2} - \mu^2$$

Model 2: Variational Inference

Model 2: Variational Inference

Idea: instead of deterministic weights, each layer models a probability distribution over parameters, capturing uncertainty via **Gaussian variational inference**.

- **Reparameterization Trick** : Enables backpropagation through sampled weights, using:

$$\theta = \mu + \sigma \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

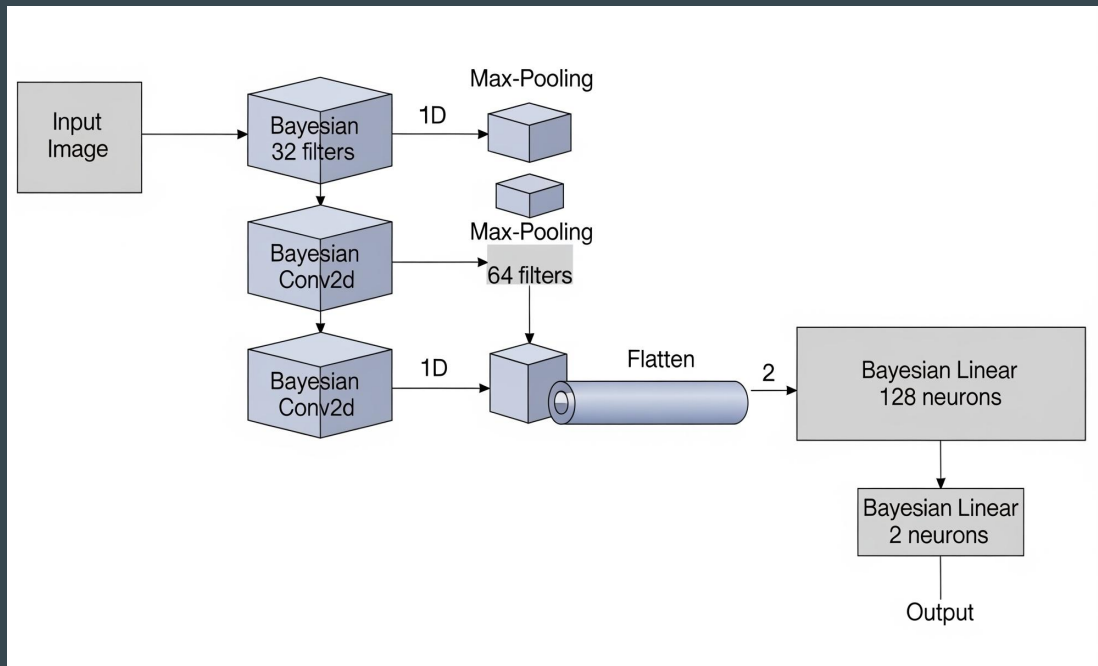
- **Softplus Activation** : Ensures positive variance in Bayesian layers to prevent learned σ from collapsing to zero:

$$\text{Softplus}(x) = \ln(1 + e^x)$$

- **Monte Carlo Sampling**: Used to approximate expectations in the ELBO during training

Architecture

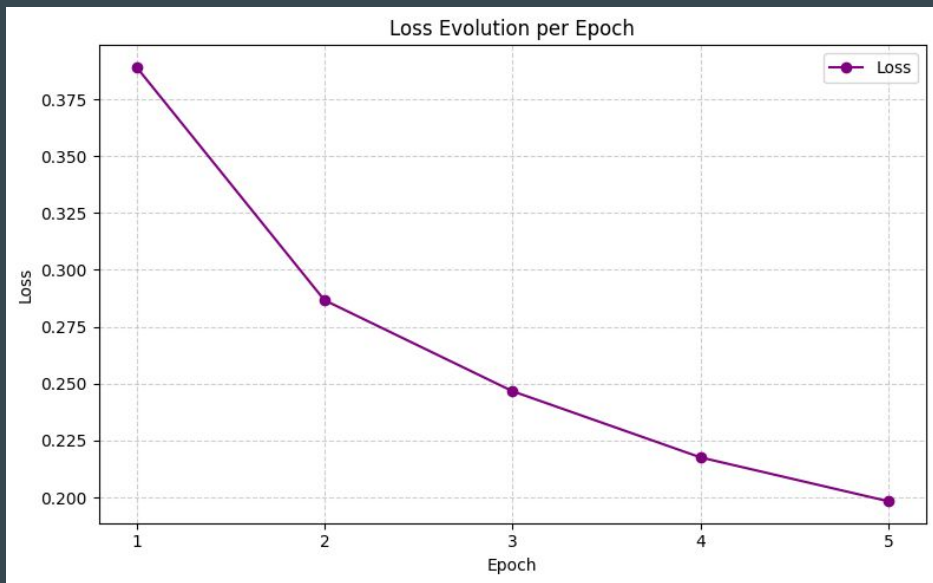
- **Input**: RGB image ($3 \times 64 \times 64$)
- **Conv1**: 32 filters \rightarrow ReLU \rightarrow MaxPool ($\rightarrow 32 \times 32 \times 32$)
- **Conv2**: 64 filters \rightarrow ReLU \rightarrow MaxPool ($\rightarrow 64 \times 16 \times 16$)
- **Flatten** \rightarrow Fully connected:
 - **FC1**: 16,384 \rightarrow 128 (ReLU)
 - **FC2**: 128 \rightarrow 2 classes (**real/fake**)



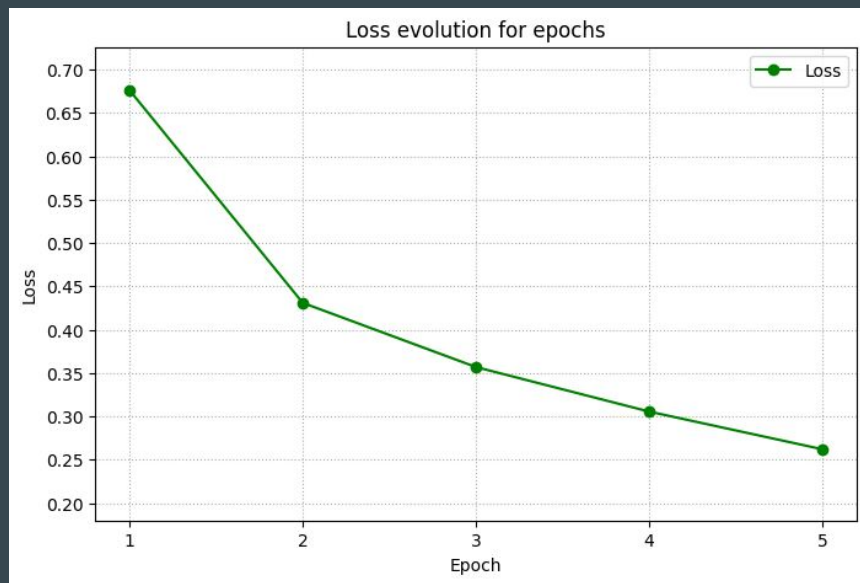
Conclusions

Training Loss

The models were trained for **five epochs**, showing distinct optimization patterns: VI starts with a higher loss but decreases sharply, while MC-Dropout follows a more gradual decline.



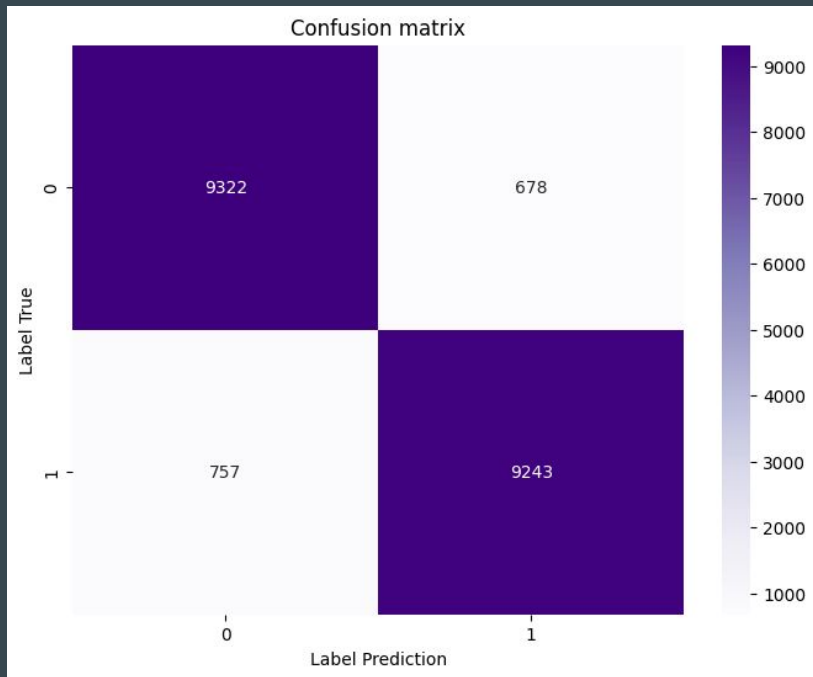
MC-DROPOUT



VI

Confusion matrix

MC-Dropout has fewer **False Positives** but more **False Negatives**, while VI exhibits the reverse trend, indicating distinct uncertainty patterns between the two models.



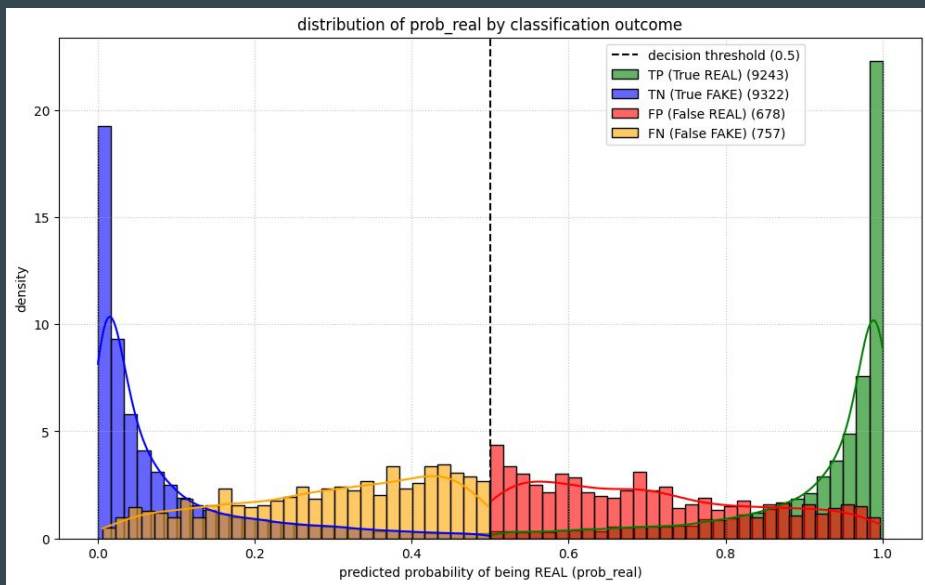
MC-DROPOUT



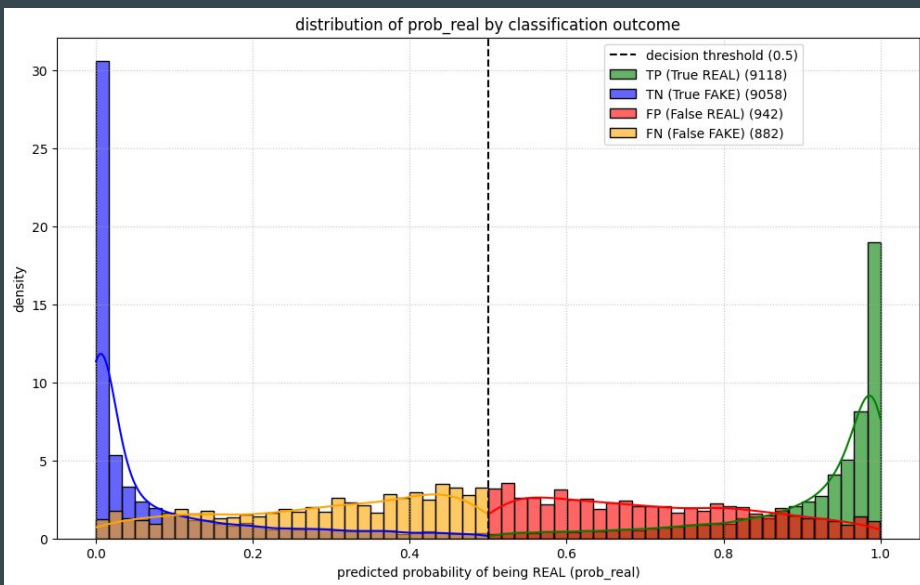
VI

Distribution of real probability

Both models show **asymmetric confidence**, correct predictions are concentrated near probabilities 0 and 1. Misclassified samples tend to cluster around 0.5, with VI exhibiting greater uncertainty, while MC-Dropout produces more extreme predictions.



MC-DROPOUT



VI

Accuracy

Class	Recall	F1-Score	Precision
REAL	0.9243	0.9280	0.9317
FAKE	0.9322	0.9285	0.9249

MC-DROPOUT

Class	Recall	F1-Score	Precision
REAL	0.9118	0.9091	0.9064
FAKE	0.9058	0.9085	0.9113

MC Dropout achieves a higher overall **accuracy** of 92.83%, outperforming the VI 90.88%, and shows more balanced classification performance, as confirmed by higher precision, recall, and F1-score.

Ultimately, MC Dropout offers **strong predictive performance with faster training convergence**, whereas Bayesian CNN provides **richer uncertainty quantification**, making it potentially more suitable for handling ambiguous or high-risk classification tasks.

Thank you!