An Improved CNN Steganalysis Architecture based on "Catalyst Kernels" and Transfer Learning

Rabii El Beji Marwa Saidi Houcemeddine Hermassi Rhouma Rhouma

May 4, 2018

Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis, LR16ES07 Robotique, Informatique et Systèmes Complexes, Tunis, Tunisia

Table of contents

Introduction to Steganograpy and Steganalysis

What's Stenography?

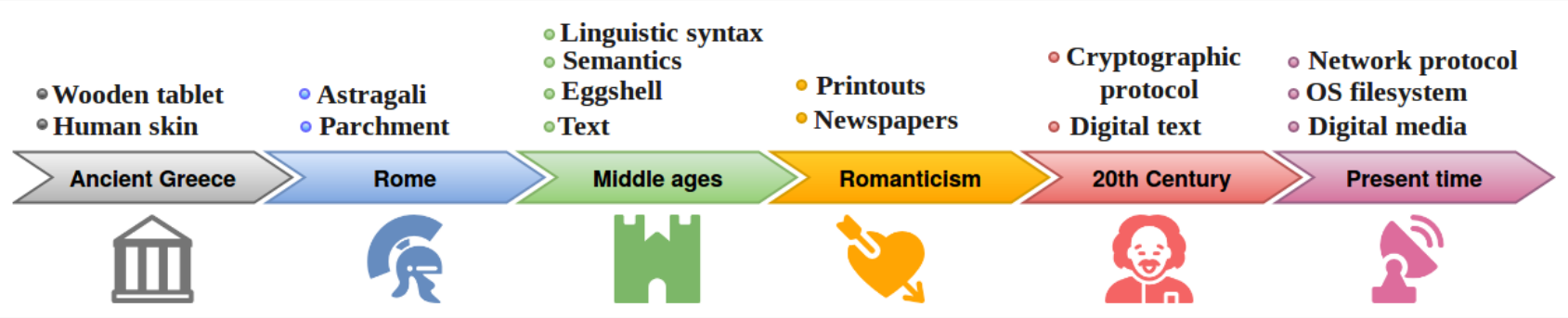
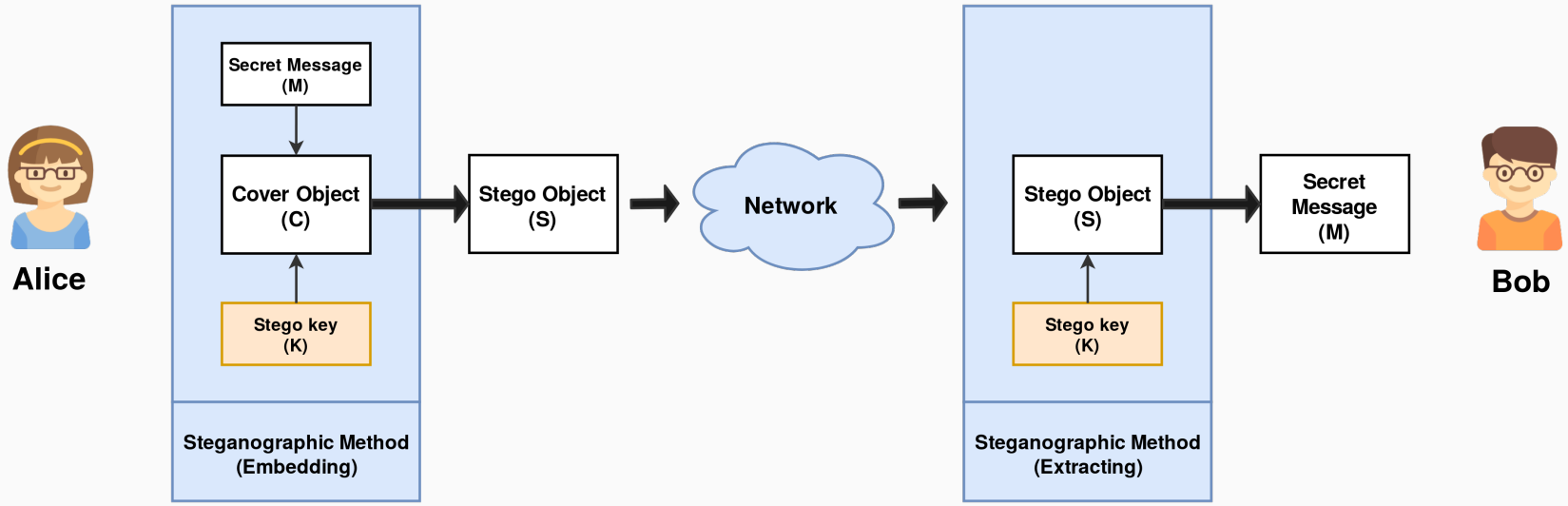


Figure: A timeline of the evolution of steganographic techniques

Recently, Four main trends of development of the so-called digital steganography can be distinguished:

- Linguistic steganography
- File system steganography
- Network steganography
- **Digital media steganography**

A Stegosystem !

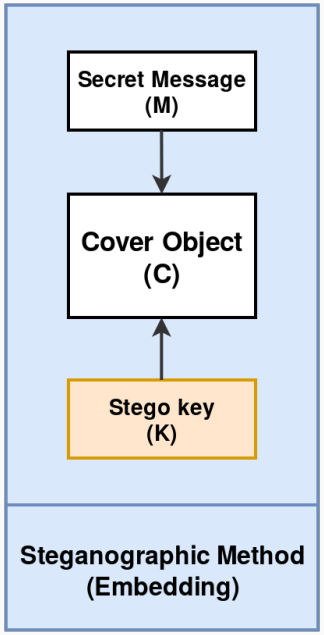


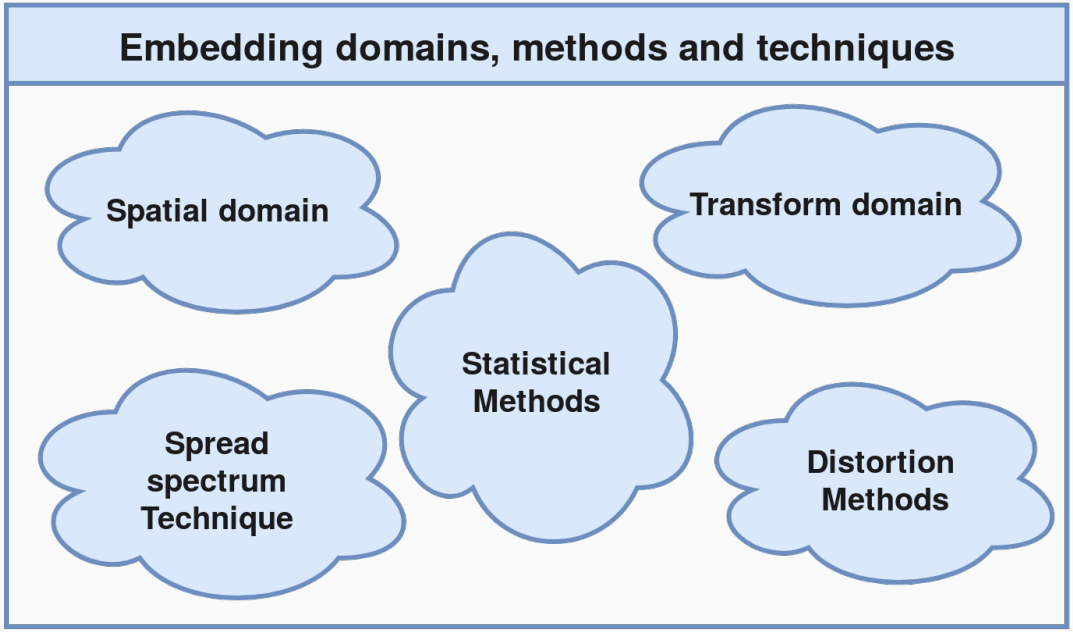
The main keywords used in the most stegosystems are:

- Cover Object
- Stego Object
- Stego Key
- Embedding Domain

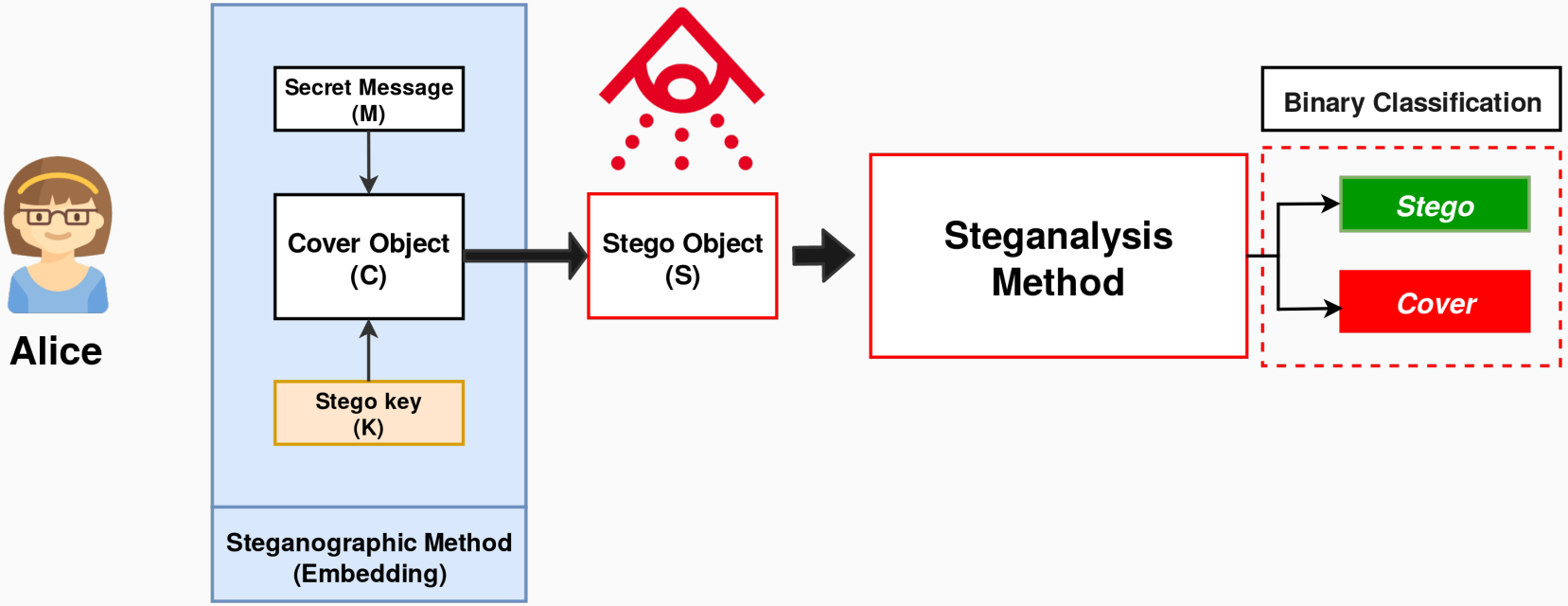
Steganographic Embedding Domains

Alice

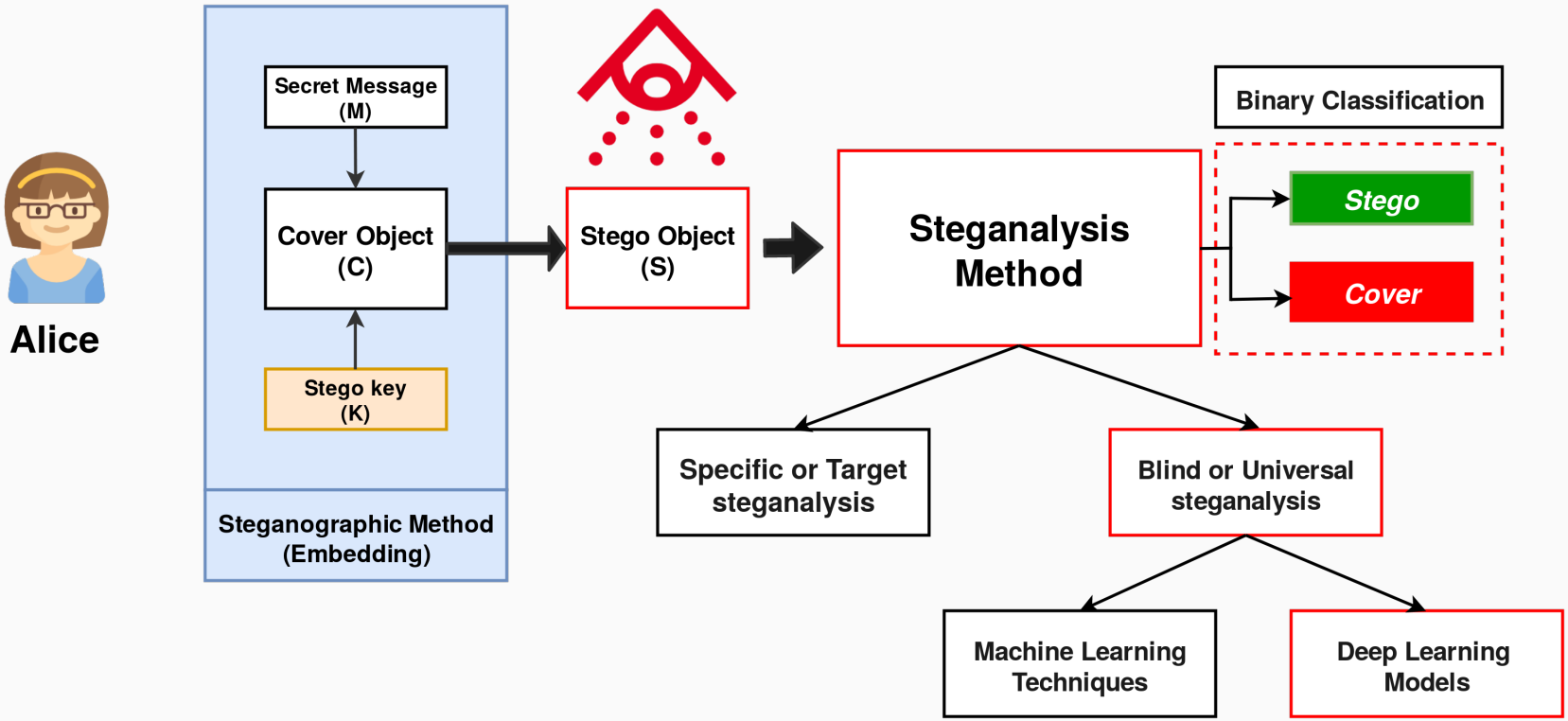




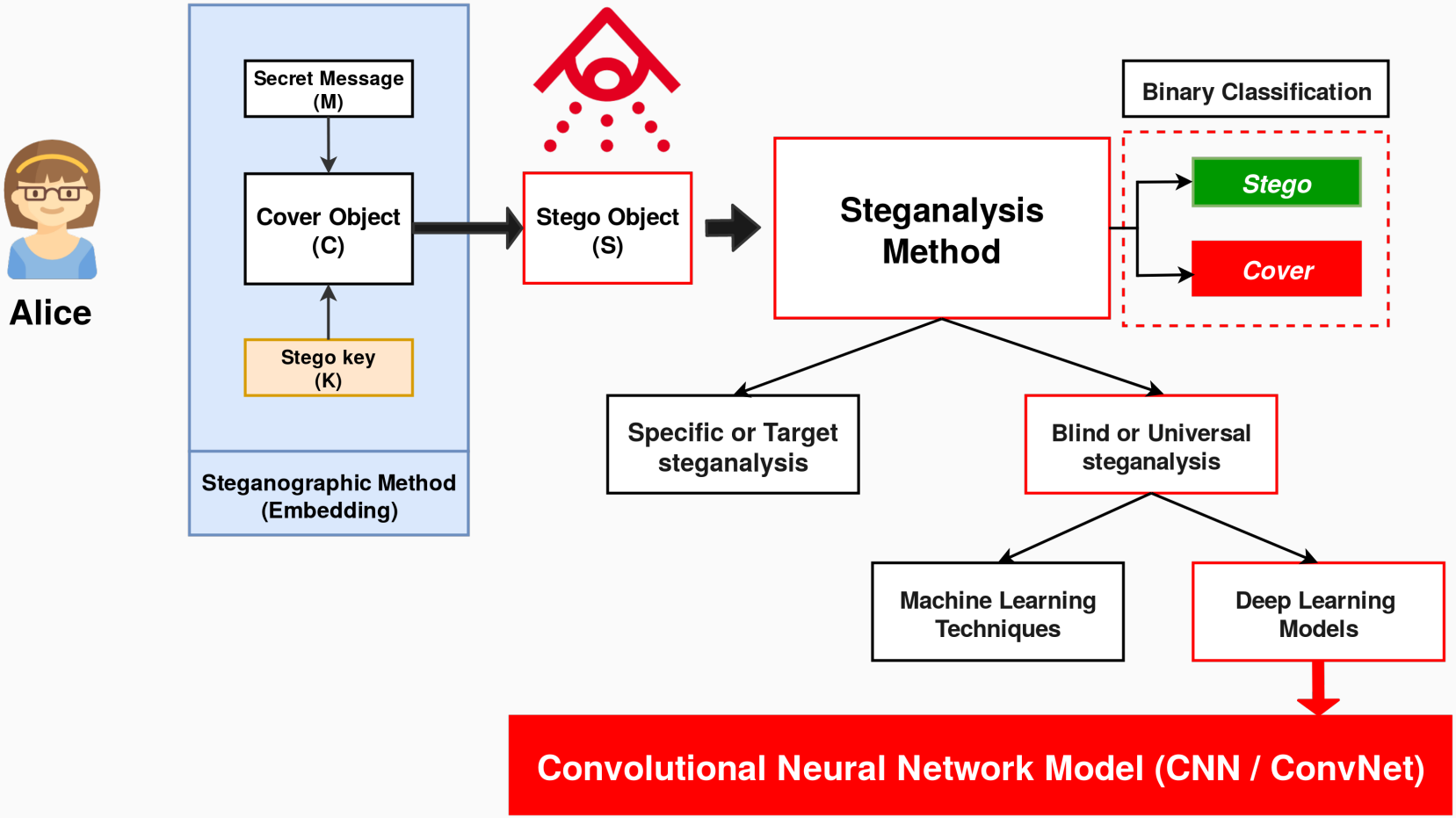
Steganalysis



steganalysis

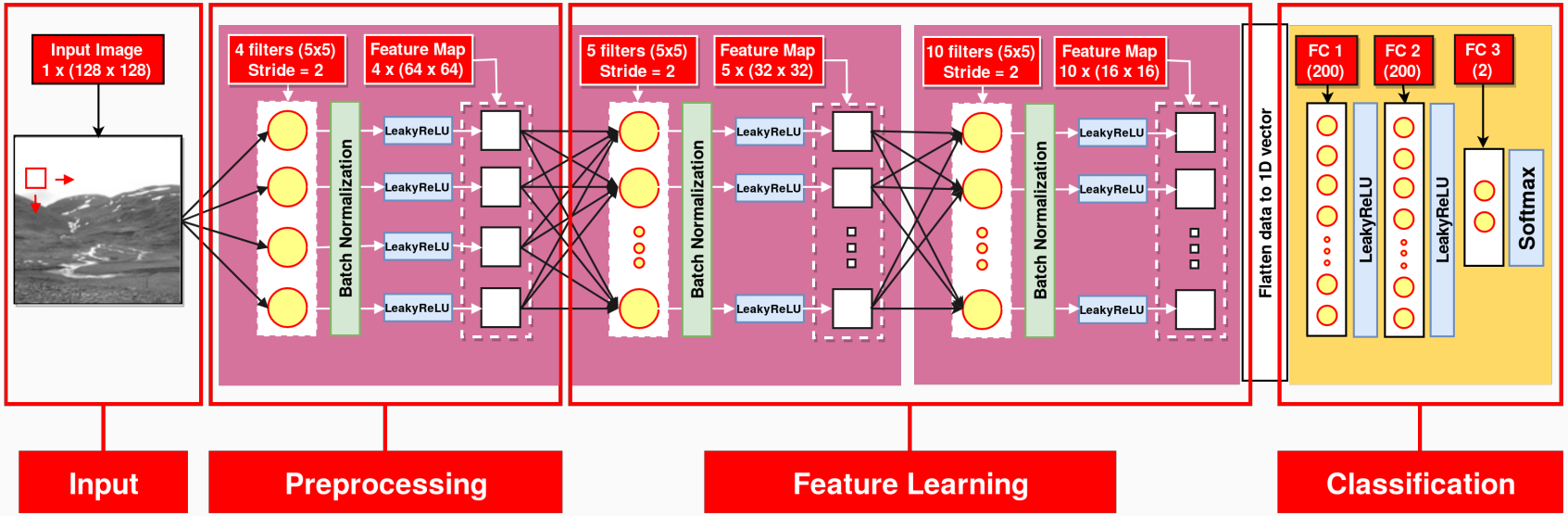


steganalysis



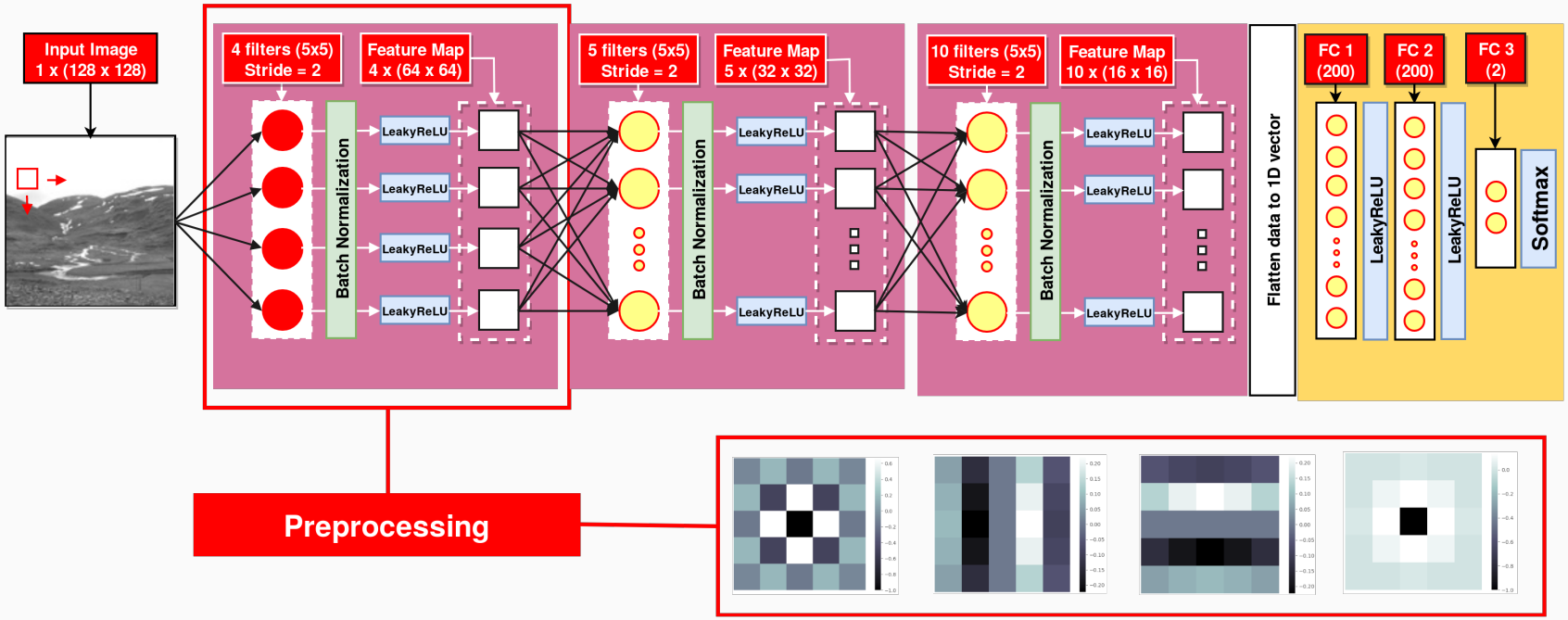
Proposed CNN Model

Proposed CNN Model



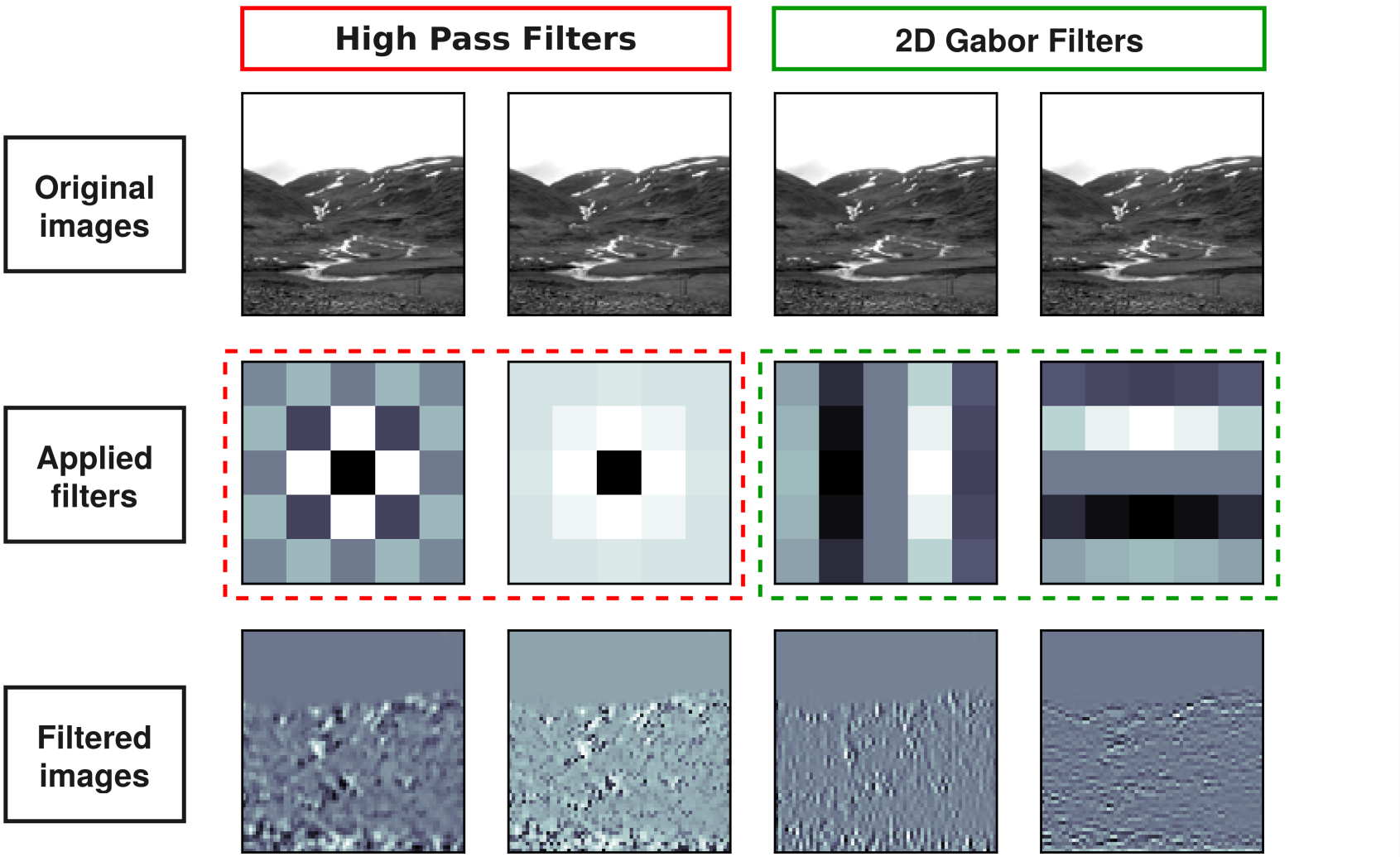
- An input image in grayscale with a size of 128×128 pixels
- Three hidden or convolution layers
  - One convolution layer for preprocessing
  - Two convolution layers for dynamic learning
- Three fully connected layers for classification + *Softmax* classifier

Preprocessing layer using "Catalyst Kernels"

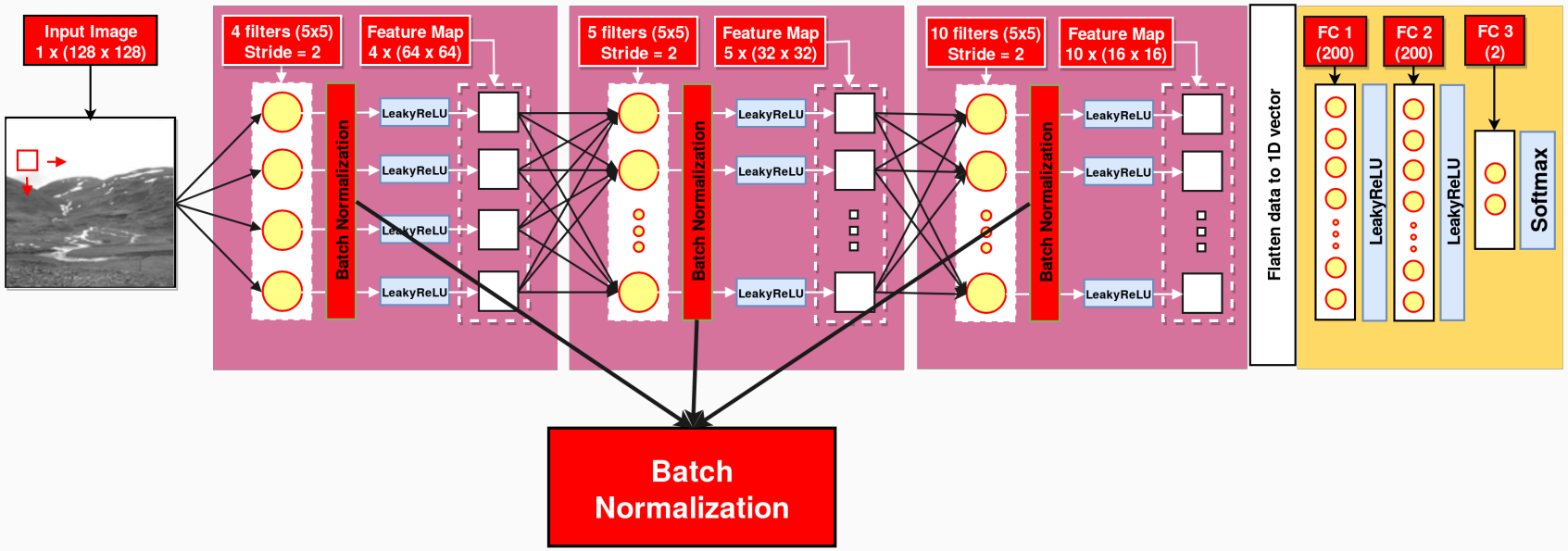


- Increase the *SNR* between the cover and the stego objects
- Reduce as much as possible the similarity between stego and cover objects

Preprocessing layer using “Catalyst Kernels”

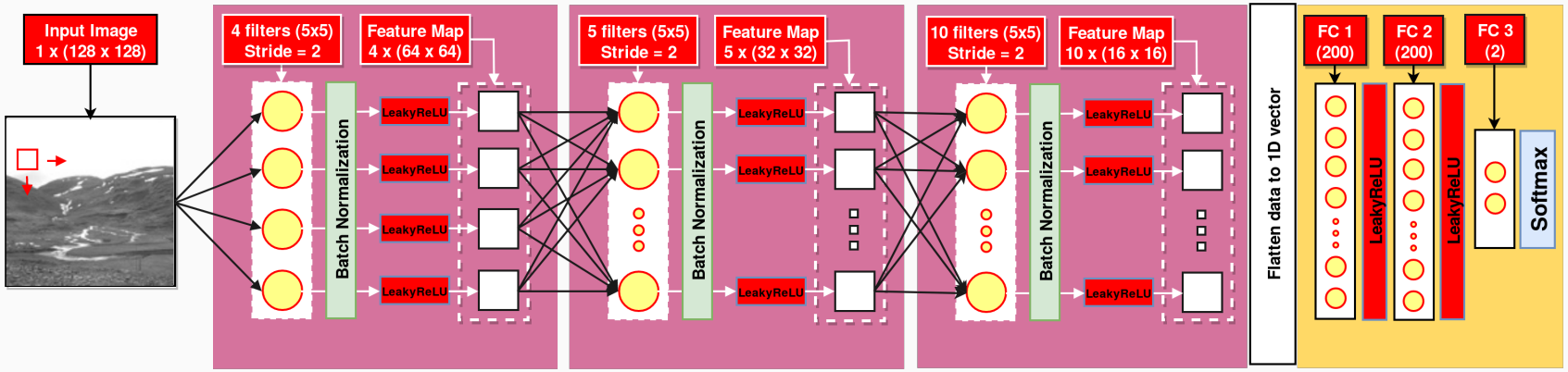


Batch Normalization

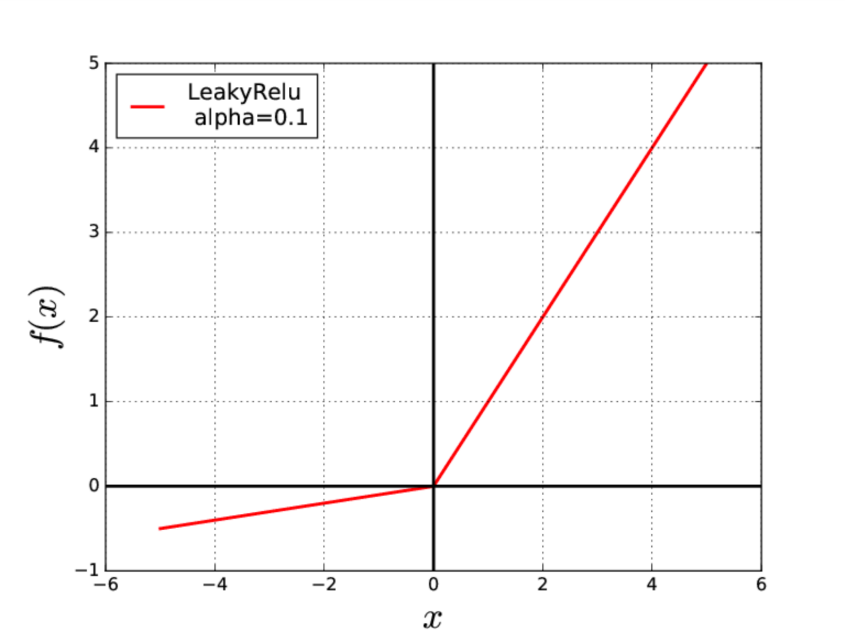


- Normalize the weak stego noise.
- Speed up the learning and improves the detection accuracy

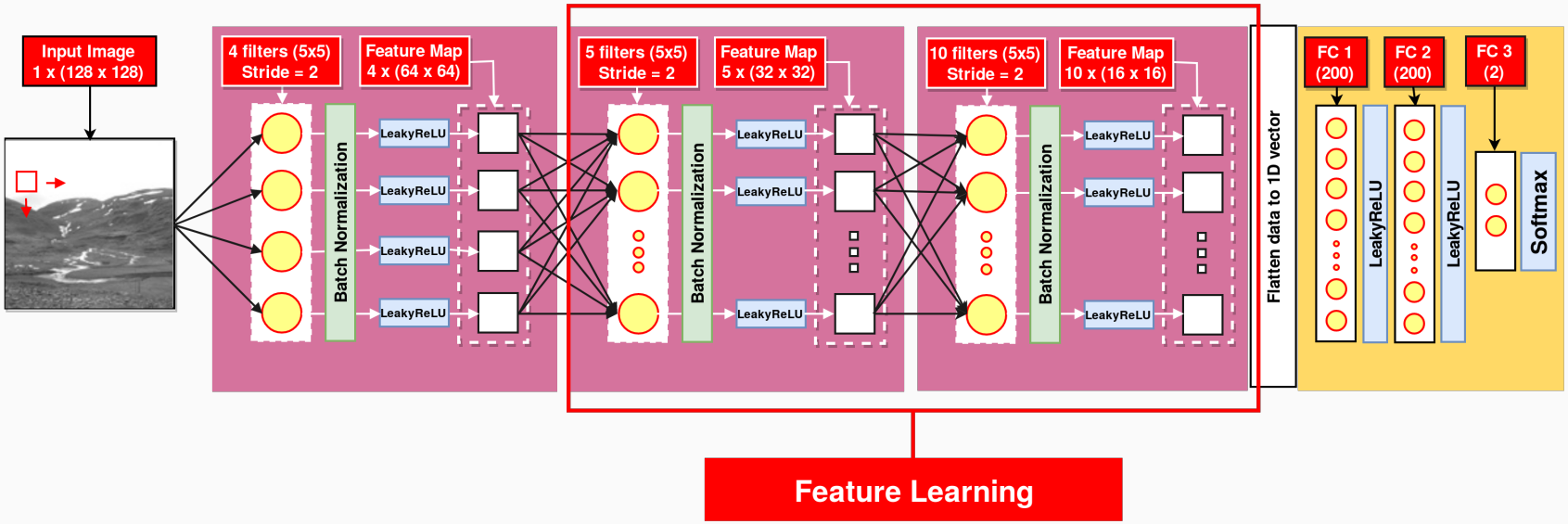
Activation Function (LeakyReLU)



Activation Function: LeakyReLU

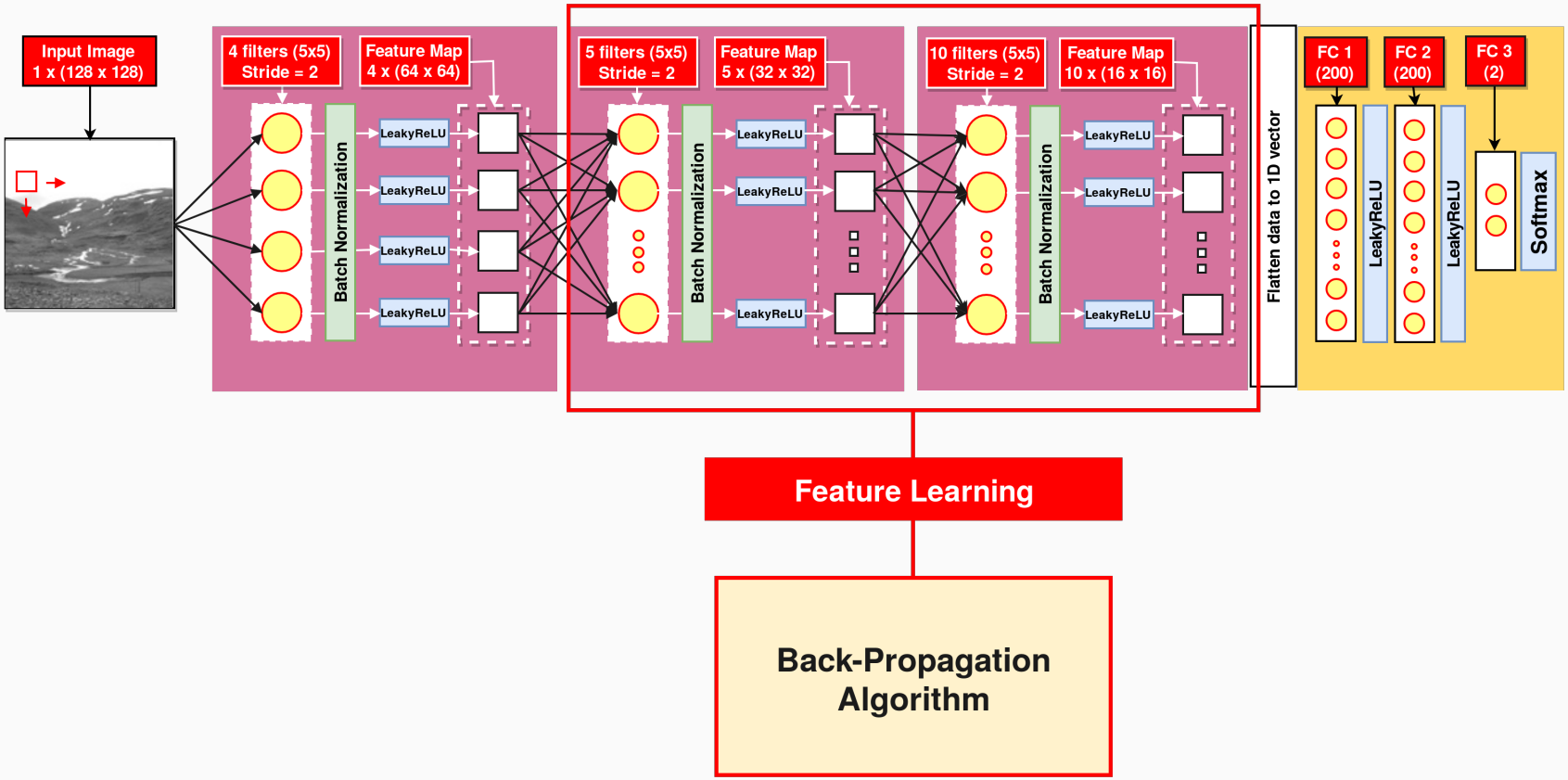


Feature Learning

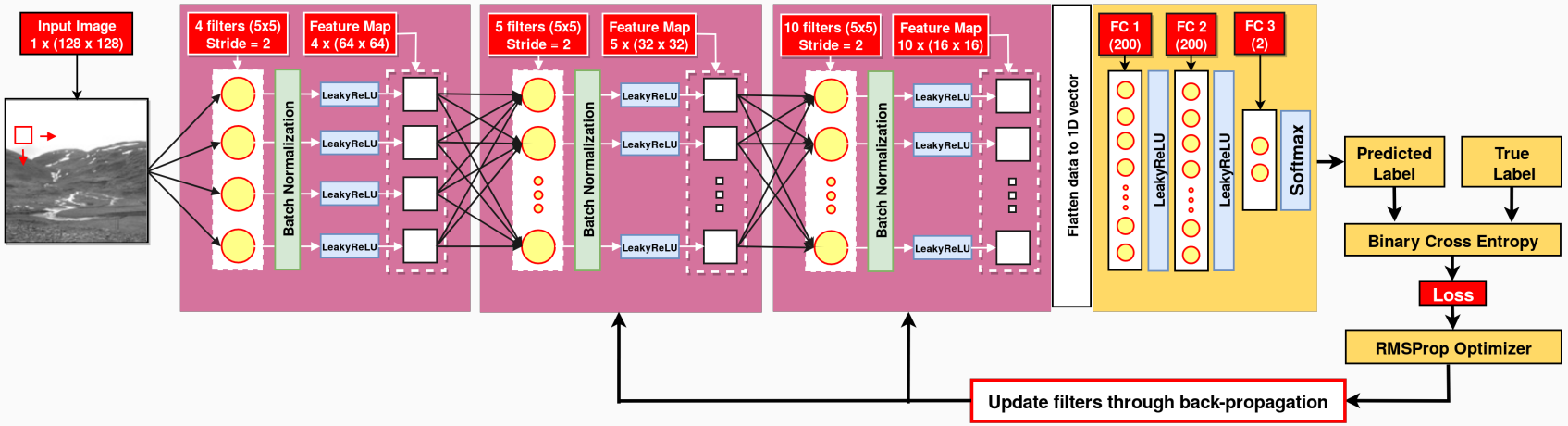


- The values of the filters will be updated dynamically during the learning process
- The update will be done at each iteration for a better separation between the two classes (stego and cover)

Feature Learning

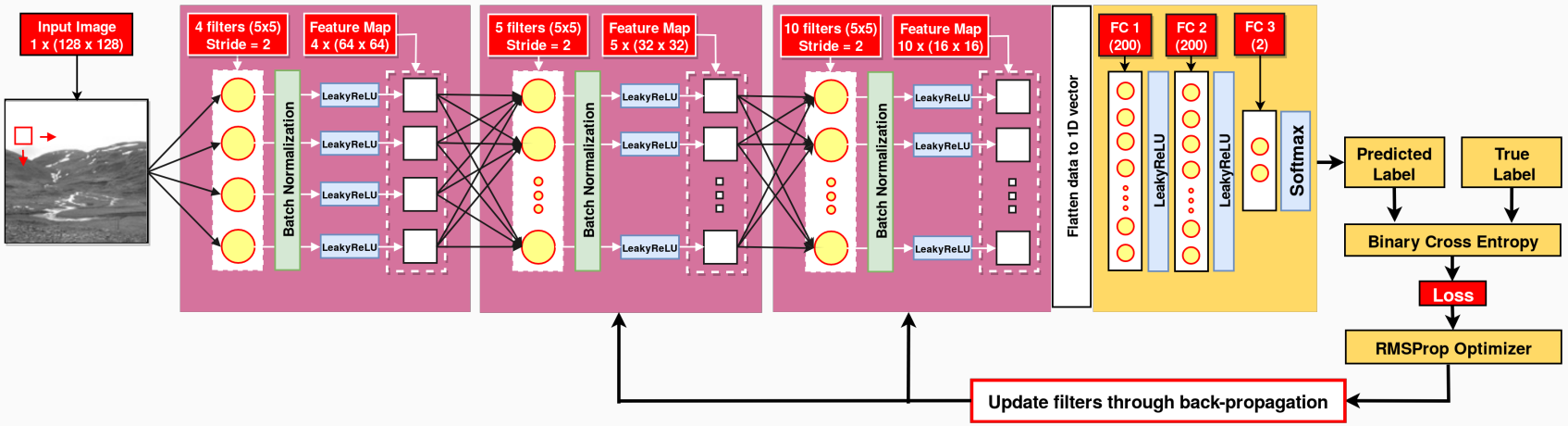


Classification & Back-Propagation Algorithm



- **Binaray Cross Entropy** : Loss function to compute the classification error.

Classification & Back-Propagation Algorithm

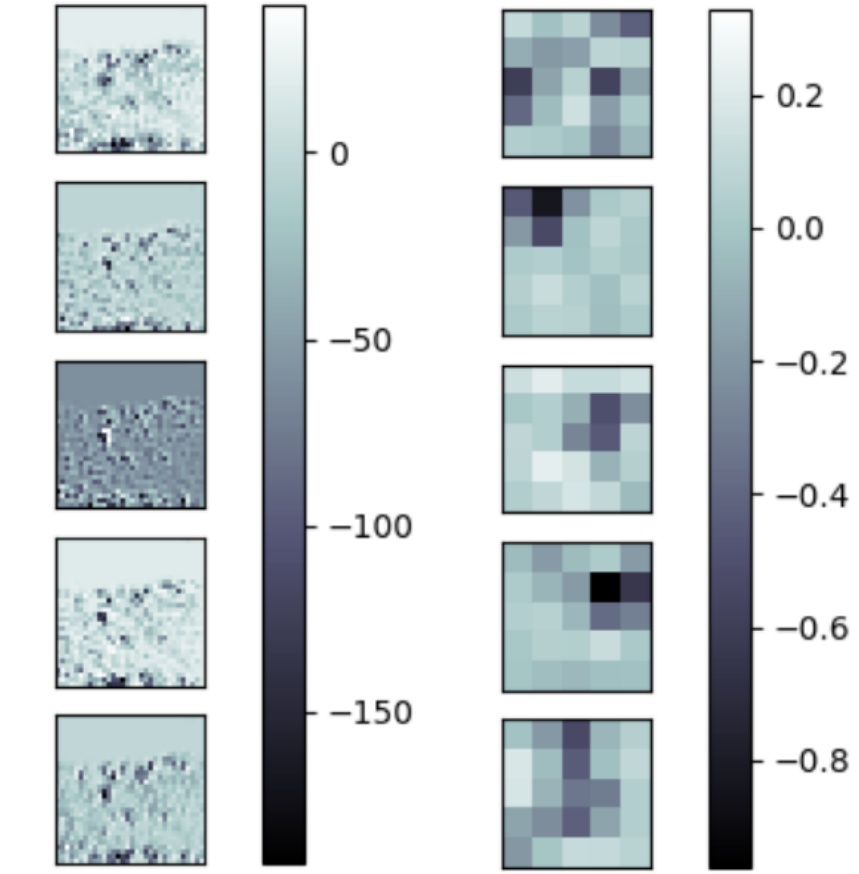


- **Binaray Cross Entropy** : Loss function to compute the classification error.
- **RMSPop Optomizer (for Root Mean Square Propagation)** : Adjust the weight value gradually basing on the back-propagation algorithm

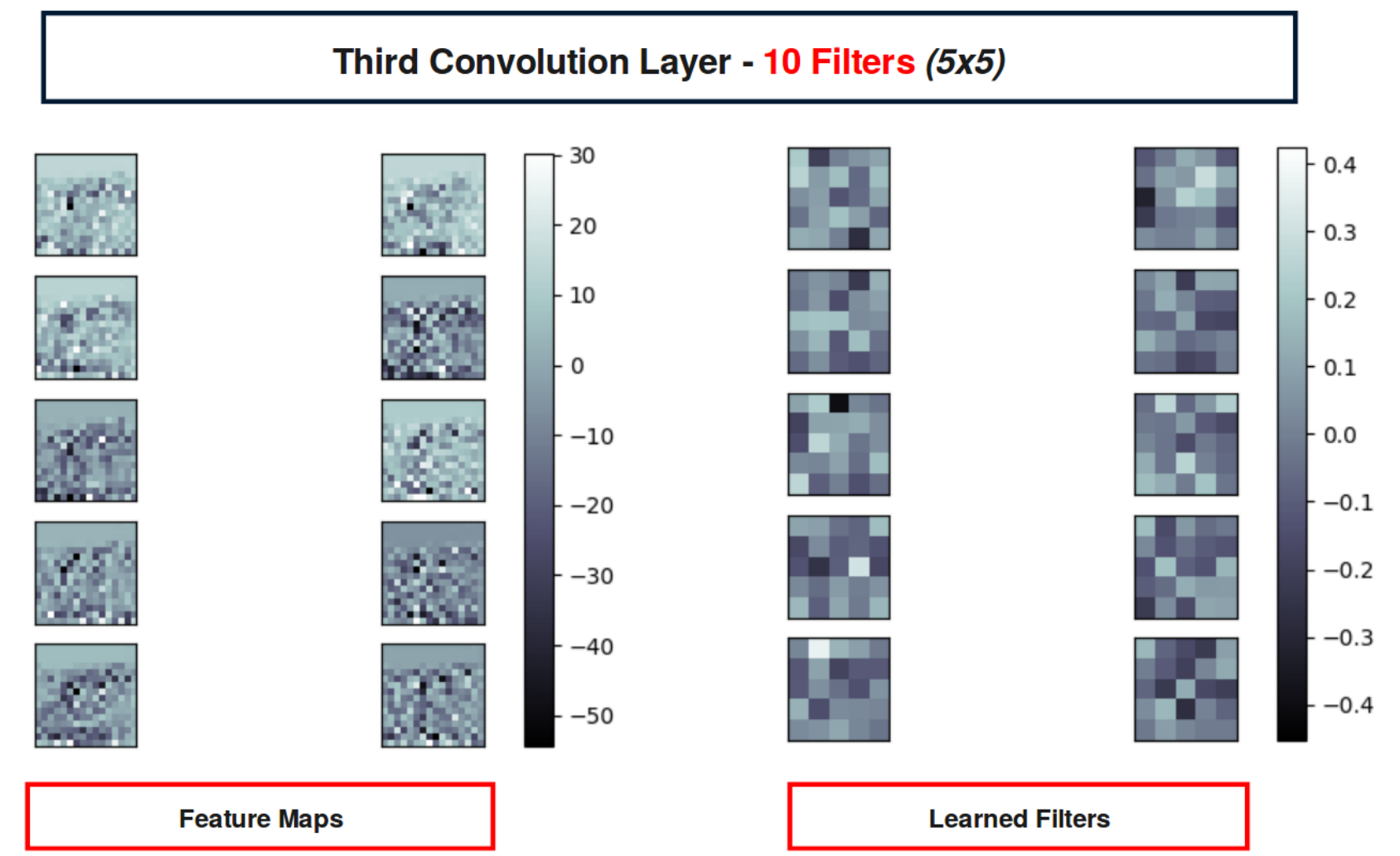
Filters and Feature Maps after learning

Second Convolution Layer

5 Filters *(5x5)*



Filters and Feature Maps after learning

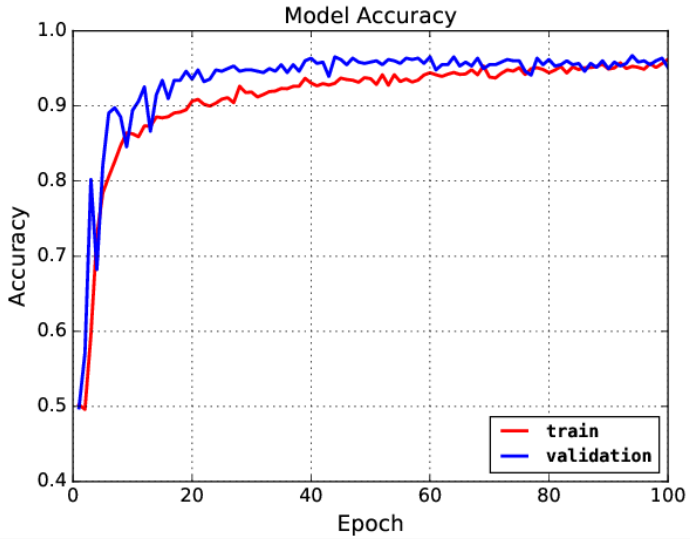


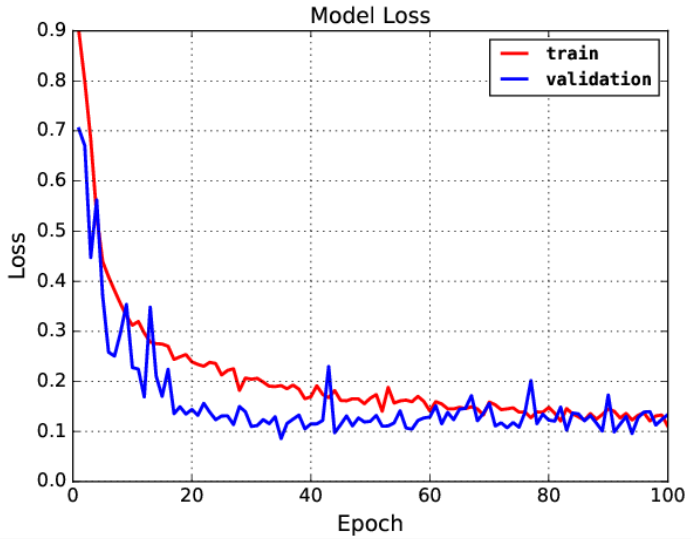
Learning from scratch and Transfer Learning

Learning from scratch

1- For high payloads [1 bpp]

- We trained our network from scratch for 100 epoch.
- Training model accuracy (right) and loss (left) using **WOW** algorithm.

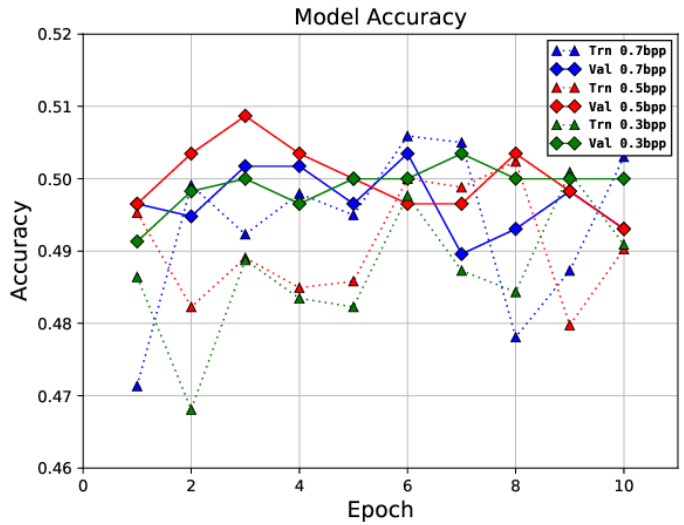


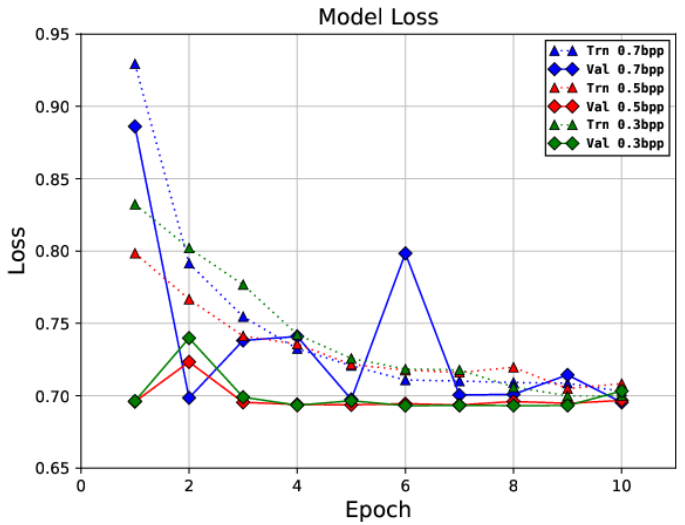


Learning from scratch

2- For low payloads [0.7, 0.5 or 0.3 bpp]

- It's impossible to ensure a convergence with our model for the first 10 epoch at least.

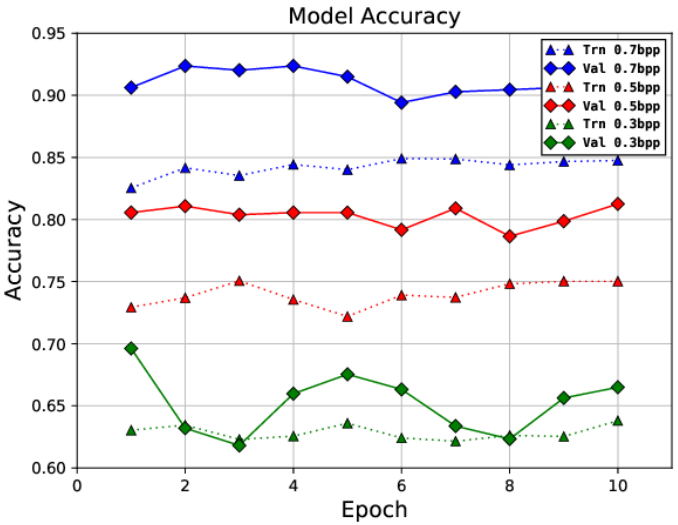


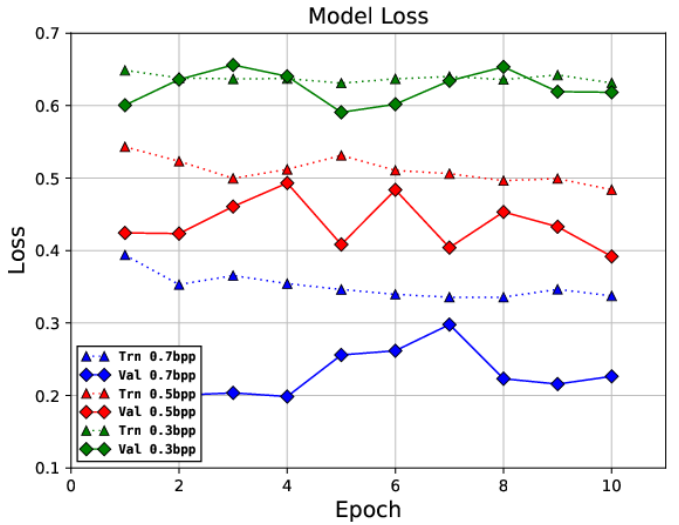


Transfer Learning

Transfer learning for low payloads (0.7, 0.5 or 0.3 bpp)

- The model show convergence from the first epoch.





Experimental setup & Results

Dataset & Implementation Environment

Dataset

- We used a chunk of the public dataset named BossBase v1.01
- We just took the first 2400 images, resized to $(128 \times 128 \times 1)$ pixels
  - 1700 images as training data.
  - 300 images as validation data.
  - 400 images as testing data.

Software platform

- The training of the CNNs is performed on a Keras API using TensorFlow as backend.

Hardware

- The implementation of the algorithms was performed on a computer CPU Intel Core $i3 - 2370M, 2.40GHz \times 4$, with Ubuntu 14.04 LTS.

Results: Model evaluation

- We choose three recent, highly undetectable stenographic algorithms in the spatial domain:
  - ***WOW*** (2012)
  - ***S-UNIWARD*** (2014)
  - ***HUGO*** (2010)
- We prepared for the testing process different datasets with (1*bpp*), (0.7*bpp*), (0.5*bpp*) and (0.3*bpp*) payloads

	S-UNIWARD				**WOW**				**HUGO**			
Payload (bpp)	1.0	0.7	0.5	0.3	1.0	0.7	0.5	0.3	1.0	0.7	0.5	0.3
Accuracy	0.96	0.93	0.88	0.73	0.96	0.92	0.83	0.67	0.96	0.91	0.84	0.69
P_E	0.04	0.07	0.12	0.27	0.04	0.08	0.17	0.33	0.04	0.09	0.16	0.31

Results: Detection errors *vs* payloads

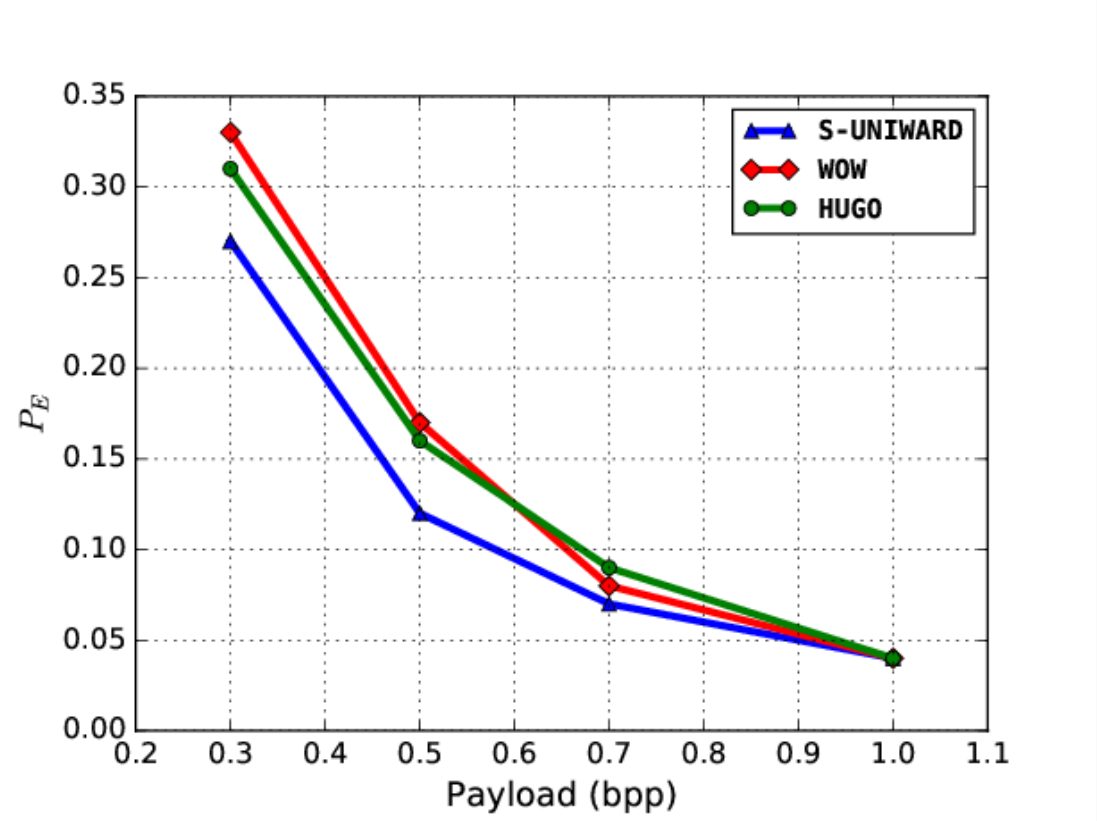
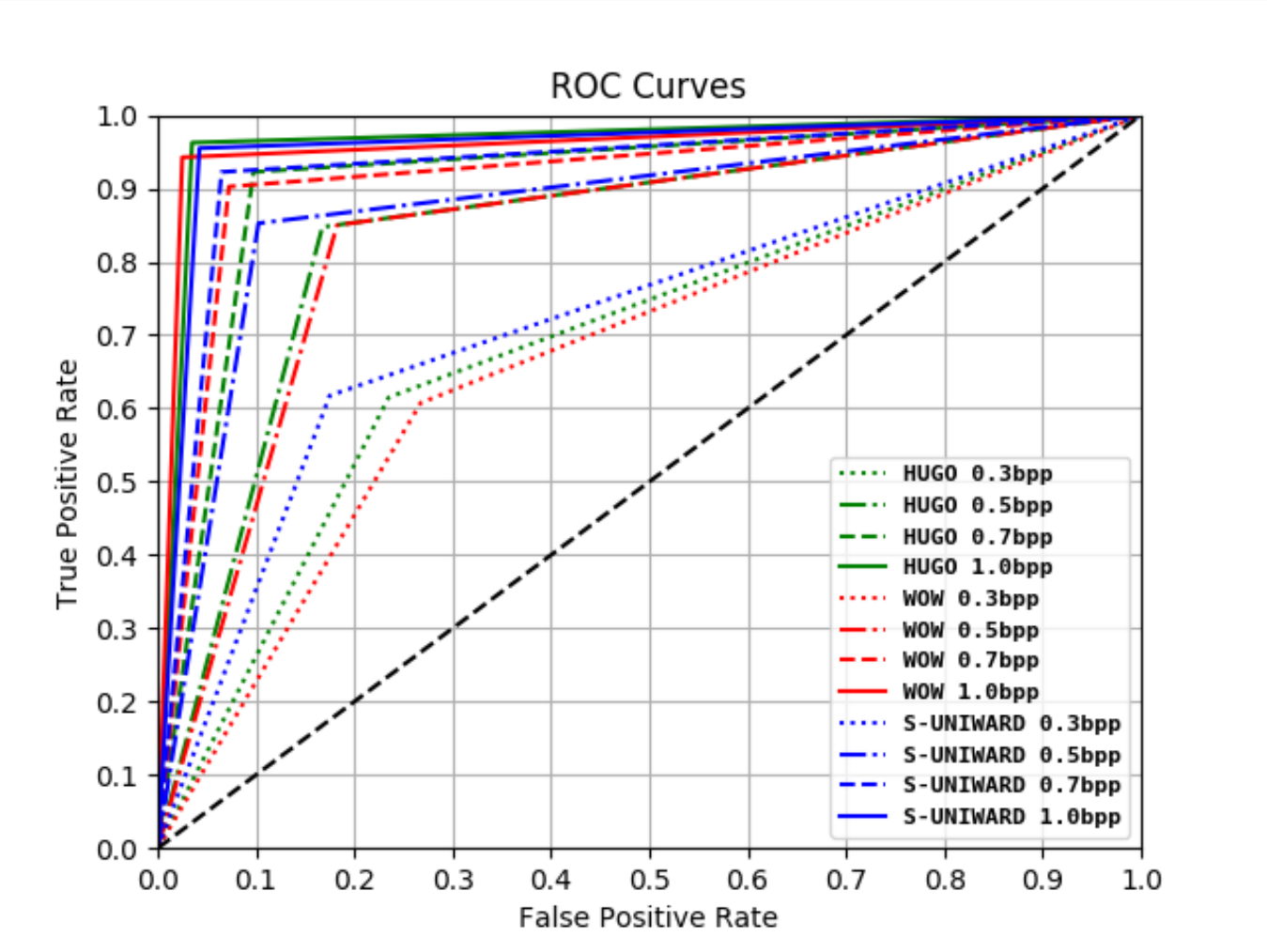


Figure: Detection errors of the proposed classifier for the 3 state-of-the-art stegano- graphic schemes (S-UNIWARD, WOW, HUGO) as a function of the embedded pay- load.

Results: ROC curves



Conclusion & Future work

Conclusion & Future works

- **Conclusion**
  - ⇒ We implemented a new CNN model for Steganalysis with small-scale images and low-cost computing resources.
  - ⇒ The CNN model showed good performance with highly undetectable steganographic algorithms
- **Future works**

⇒ We will focus on the concept of multi-class steganalysis using a system with multiple CNN models for predictions

Thank you for your attention!